SICHERES ÜBERGANGSLOSES ROAMING

Vom Fachbereich Informatik
der Technischen Universität Darmstadt
zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften (Dr. rer. nat.)
genehmigte Dissertation
von

Dipl.-Wi.-Ing. Dipl.-Inform. Michael Haisch

aus
Saarburg

Referenten: Prof. Dr. rer. nat. Claudia Eckert
Prof. Dr.-Ing. Ralf Steinmetz

Tag der Einreichung: 2.2.2007
Tag der mündlichen Prüfung: 2.7.2007

Darmstadt 2007
D17

Summary

In this work an architecture for secure seamless roaming has been developed allowing seamless roaming from every place in the world using distinct network resources and access networks of distinct providers. At first a review of today's communication techniques is shown to prove their applicability to this paradigm and similar projects are referenced. The involved parties are identified and then the interests of these parties are analysed. Generic business models are developed taking parties' interests and their relationships into account.

In the business models the swap to networks of different providers is specified as an isolated service. Trust models are deduced from the business models and claims to the architecture with regard to the security provided for are described. Especially secure mutual authentication creating provable data of accounting and authorisation are depicted. Further on an important feature of the architecture is the irreducible number of the user's interactions for the authentication caused by Single Sign On.

Furthermore it is investigated how to implement these approaches using communication techniques. Two prototypes are implemented providing for secure mutual authentication. The first implementation does mutual authentication between a client and a server und the second uses the communication via "Wireless Local Area Networks" (WLAN) and "Worldwide Interoperability for Microwave Access" (WiMAX). The tests of the implementations are operating successfully.

Finally assumptions of trends in future and suggestions are launched.

Summarised it is shown theoretically and practically how based on the results of the approaches of the business models seamless security roaming between heterogeneous networks of different domains works.

Eidesstattliche Erklärung

Hiermit versichere ich an Eides Statt, dass ich die vorliegende Arbeit selbständig und nur unter Benutzung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Darmstadt, den 2. Februar 2007

Michael Haisch

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Das Verlangen, möglichst jederzeit Zugriff zum Internet zu haben, um zum Beispiel "surfen" oder "chatten" zu können, kurz immer„online“ (always on) zu sein, hat in den letzten Jahren stark zugenommen und wird auch in Zukunft noch weiter zunehmen. Dies liegt zum einen daran, dass die Zahl der Dienste, die für verschiedenste Nutzer bereitgestellt werden, schon jetzt sehr hoch ist und auch in Zukunft weiter ein Ansteigen zu erwarten ist. Zum anderen nimmt auch die Anzahl und Vielfalt der Endgeräte zu, welche auf die Bedürfnisse der unterschiedlichsten Nutzer zugeschnitten sind. Handys, Laptops, Personal Digital Assistants und Smartphones sind neben Personal Computers eine Selbstverständlichkeit für den modernen Menschen des dritten Jahrtausends. Insbesondere Unternehmen haben zunehmend ein Interesse daran, ihren Mitarbeitern das Arbeiten mit dem entsprechenden Equipment über das Internet zu ermöglichen. Dabei gilt die Aussage: „Der quantitative Nutzen misst sich in Kostensenkung oder neuen Umsatzquellen. Der Qualitative Effekt besteht aus besserem Kundenservice und einer effizienteren Zusammenarbeit innerhalb der Organisation.“ [CZ06]. In diesem Statement ist der angestrebte Fortschritt treffend zusammengefasst.

Um diese Endgeräte nutzen zu können, muss man Zugang zum Internet haben. Man erhält diesen über verschiedene Zugangsnetze. Im Einzelnen sind dies unterschiedliche Mobilfunknetze der zweiten Generation, wie GSM/GPRS[1] und der dritten Generation, wie UMTS[2], und WLAN[3]s in so genannte „Hot Spots“ z. B. in öffentlichen Cafés oder Flughäfen. Mitarbeiter von Unternehmen können so von unterwegs oder auch zu Hause auf Firmendaten zugreifen, wie wenn sie sich an ihrem Arbeitsplatz im Unternehmen vor Ort befinden.

Diese Zugänge zum Internet sind nicht frei verfügbar, sondern sie werden vertraglich mit den Betreibern der Zugangsnetze und mit den Internet-Dienst-Anbietern, Internet Service Providern, geregelt. Zurzeit kann man zwar von überall auf der Welt Zugang zum Internet, zum eigenen Firmennetz bzw. Heimnetz zu haben, aber man muss verschiedene Verträge mit unterschiedlichen Internet Service Providern oder mehreren Bereitstellern von Zugangsnetzen abschließen. Ein Wechsel zwischen den heterogenen Netzen kann nicht vollzogen werden, ohne dass eine erneute Authentifikation für den Nutzer notwendig wird und ohne dass die Verbindung beim Wechsel zwischen den unterschiedlichen Netzen abreist. Ein bestimmtes gewünschtes Sicherheitsniveau bleibt bei diesen Vorgängen jedoch nicht aufrechterhalten. Dies ist die Realität zum gegenwärtigen Zeitpunkt.

[1] Global System for Mobile Communication (GSM) / General Packet Radio Service (GPRS)

[2] Universal Mobile Telecommunication System (UMTS)

[3] Wireless Local Area Network (WLAN)

Momentan besteht zwar die Möglichkeit z. B. über UMTS eine Internetverbindung zu erhalten und prinzipiell wäre es möglich, mit der UMTS Technologie allein flächendeckend weltweit ein übergangsloses Roaming zu ermöglichen.

Da der Begriff des Roaming nicht einheitlich verwendet wird, erfolgt hier zunächst eine Klärung desselben: Roaming beinhaltet in dieser Arbeit nach [Schiller03] den Wechsel zwischen Netzen unterschiedlicher Betreiber. Damit wird Roaming als eine besondere Art des Handovers bzw. Handoffs bezeichnet. Übergaben zwischen zwei Funkzellen derselben Technologie werden im Gegensatz dazu als horizontaler Handover bezeichnet. Wechselt man zwischen Zellen unterschiedlicher Technologie spricht man von einem vertikalen Handover.

Ein auf UMTS alleine beruhendes Roaming würde die Zusammenarbeit der unterschiedlichen in der Welt etablierten Mobilfunkanbieter und eine wirklich flächendeckende über Ballungsgebiete und Großstädte hinausgehende UMTS Versorgung voraussetzen. Dies ist bis in der Realität jedoch nicht der Fall, und selbst, wenn dies gegeben wäre, wäre ein solches System ineffizient, da es vergleichsweise teuer wäre. Es gibt ein großes Spektrum kostengünstigerer drahtloser Übertragungstechnologien, wie zum Beispiel WLAN, eine Vielzahl Mobilfunkanbieter und verschiedene Internetdienstanbieter. Diese alle müssen in ein System integriert werden, das es einem Nutzer ermöglicht, während er irgendwelche Dienste nutzt, weltweit zwischen den verschiedensten Zugangsnetzen und Internet-Dienst-Anbietern transparent wechseln zu können.

Bisher gibt es lediglich Insellösungen, wie die z. B. die Deutsche Telekom und die schweizerische Telekom anbieten. Beide ermöglichen es, zwischen auf unterschiedlichen Funktechnologien beruhenden Netzen desselben Betreibers zu wechseln, an. Der Stand der Technik beschränkt sich gegenwärtig allerdings auf Lösungen, die ein Handover zwischen WLAN als lokalem Netz und UMTS/GSM/GPRS als Mobilfunknetze der zweiten und dritten Generation innerhalb eines Anbieters ermöglichen. Swisscom Mobile hat bereits im Jahr 2004 eine PCMCIA-Combo-Karte angeboten welche automatisch das schnellste verfügbare Netz sucht und übergangslos einen Wechsel vollzieht. Diese Möglichkeit des Wechselns beschränkt sich allerdings auf die Netze der Swisscom [http://www.3g.co.uk/PR/June2004/7873.htm]. T-Mobile bietet unter dem Stichwort „Web`n`walk“ eine PCMCIA[4] Karte für GPRS/UMTS/WLAN an, die den Wechsel von UMTS zu Telekom eigenen WLAN „Hot Spots“ ab Sommer 2006 auch unterbrechungsfrei ermöglicht. Ein Wechsel zu WLANs von anderen Betreibern mit anderen Domänen ist jedoch nicht möglich.

Die Vision des „always on“ lässt sich mit den gegenwärtigen Lösungen nur gewährleisten, wenn ein Mobilfunkbetreiber flächendeckend über die gesamte Erde seine Zugangsnetze positioniert und zu einem weitgehend homogenen Netz vereint. Diese Vorstellung ist jedoch unrealistisch. Man muss davon ausgehen dass es auch in Zukunft

[4] Personal Computer Memory Card International Association (PCMCIA)

unterschiedliche Anbieter geben wird, die zueinander in Konkurrenz stehen. Daher muss eine Lösung gefunden werden, die die Interessen unterschiedlicher zueinander in Konkurrenz stehender Betreiber berücksichtigt. Dies gilt sowohl für konkurrierende Zugangsnetzbetreiber als auch Internetzugangsdienstanbieter.

Des Weiteren entspricht der Wert von Netzen heutzutage im Wesentlichen immer noch ihrer Fähigkeit Daten zu Übertragen, also der maximal möglichen Datenübertragungsrate. Obwohl die Netze der dritten Generation noch nicht flächendeckend und in breiter Masse im Einsatz sind, sind schon die Netze der vierten Generation - zumindest was Forschung angeht - auf den Weg gebracht [EITO03]. Im Allgemeinen wird jedoch angenommen, dass in Zukunft ein Paradigmenwechsel von statten geht. Der Wert der Netze der Zukunft wird nicht mehr nur von der möglichen Datenübertragungsrate bestimmt, sondern wesentlich auch von den in Ihnen angebotenen Diensten.

Aus Nutzersicht müssen diese Dienste verschiedene Anwendungen mit einem hohen Mehrwert durch neue Funktionalität, höhere Verbraucherfreundlichkeit und einen vergleichsweise höhere Sicherheitsstufe ermöglichen. Solche Anwendungen könnten entweder die Dienste eines einzigen Dienstanbieters oder die Kombination von Diensten verschiedener Anbieter beinhalten. Im Roaming Kontext impliziert dies die zusätzliche spezielle Anforderung eines möglichen „Single Sign On“, kurz SSO, da ohne ein solches SSO eine unterbrechungsfreie Dienstenutzung nicht möglich ist.

In dieser Arbeit wird deshalb eine Architektur entwickelt, welche das übergangslose oder „seamless“ Roaming zwischen heterogenen Netzen aus unterschiedlichen Domänen ermöglicht. Das bedeutet die Möglichkeit des Roaming auch zwischen Netzen unterschiedlicher Betreiber, die sich gegenseitig nicht einmal kennen. Mobile Endgeräte, wie unter anderem Handys oder Personal Digital Assistants, welche in der Regel. beim Roaming eingesetzt werden, verfügen oftmals über vergleichsweise begrenzte Ressourcen, d. h. über eine verhältnismäßig niedrige Rechenleistung, geringen Speicherplatz und eine begrenzte Bandbreite. Dies muss bei einer entsprechenden Architektur berücksichtigt werden. Die Architektur soll möglichst aus Standardkomponenten bzw. aus auf Standards bestehenden Lösungen aufgebaut sein. Die Architektur berücksichtigt zunächst die Sicherheit der Verbindungsdienste beim Zugang zu den heterogenen Netzen. Darauf aufbauend wird der wachsenden Bedeutung der Anwendungsdienste Rechnung dadurch getragen, dass die Autorisierung, welche im Rahmen der Architektur vorgeschlagen wird, es ermöglicht sicher zu stellen, dass nur autorisierte Nutzer bestimmte Dienste Nutzen können.

Die Arbeit ist folgendermaßen aufgebaut: Nach dieser Einleitung wird im nächsten Kapitel 2 ein Überblick über die grundlegenden möglichen verwendbaren Technologien erstellt. Es werden die grundlegenden drahtlosen Übertragungstechnologien hinsichtlich ihrer Eignung als Zugangsnetze und ihren Sicherheitseigenschaften sowie die für ein sicheres Roaming wichtigsten Protokolle und das bereits existierende 3GPP Interworking-Modell betrachtet.

Die Interessen der einzelnen Parteien, welche beim Vorgang des Roaming alle involviert sind, werden in Kapitel 3 analysiert. Hierfür werden generische Geschäftsmodelle entwickelt, welche die Beziehungen und Interessen der beteiligten Parteien berücksichtigen. In diesen Modellen wird der Wechsel zwischen Netzen unterschiedlicher Betreiber als isolierter Dienst angeboten. Die Vertrauensbeziehungen zwischen den unterschiedlichen Parteien werden u. a. aus den Geschäftsmodellen abgeleitet. Die wesentlichen Anforderungen an die Architektur insbesondere im Hinblick auf die zu gewährleistende Sicherheit werden analysiert und beschrieben.

In Kapitel 4 wird zunächst die grundlegende Roaming Architektur dargestellt. In Kapitel 5 wird eine generische Ideallösung für die Authentifizierung innerhalb der Roaming Architektur entwickelt. Dabei werden alle möglichen realistischen Fälle untersucht und beschrieben. Es wird danach untersucht, in wieweit sich die Architektur im Zusammenspiel von Standardlösungen und existierenden Technologien umsetzen lässt. Die Umsetzung wird detailliert beschrieben. Hierfür werden in Kapitel 6 mehrere Public Key Infrastrukturen eingesetzt. Die entwickelte Architektur beinhaltet einen zertifikatebasierten Authentifikationsprozess im Gegensatz zu der im Moment üblichen SIM[5]-basierten Authentifikation der Mobilfunkbetreiber. Stand der Technik wäre bei einer Autorisierung gegenwärtig eine auf dem Diameter Protokoll beruhende Lösung. Die hier in Kapitel 7 entwickelte Architektur beinhaltet eine auf SAML[6] basierende Autorisierungslösung, da eine auf „Diameter“ beruhende Lösung durch die gegebenen Beschränkungen des Protokolls den Anforderungen des Einsatzes bei der hier entwickelten Architektur nicht genügt. Die oben erwähnte Anforderung des „Single Sign On“ kann ebenfalls erfüllt werden. Die diesbezügliche Lösung ist in Kapitel 8 dargestellt.

In Kapitel 9 werden beispielhaft zwei Prototypen für die sichere beidseitige Authentifizierung implementiert. Zum einen ein Client und ein Server, die miteinander das auf TLS/SSL[7] basierende beidseitig sichere Authentifikationsprotokoll abwickeln und zum anderen ein zweiter Prototyp, der spezifisch für die Kommunikation über WLAN und WiMAX[8] bzw. Ethernet das Protokoll in Kombination mit einer portbasierten Zugangskontrolle nach dem Standard IEEE 802.1x abwickelt. Beide Prototypen beinhalten die notwendigen neuen Architekturkomponenten. Das implementierte Authentifikationsprotokoll wird allgemein im Zusammenspiel mit den Zertifikaten der implementierten Certification Authorities auf seine korrekte Funktion getestet. Hierfür wurden die für die in Kapitel 4 beschriebene Roaming-Architektur erforderlichen Public Key Infrastrukturen aufgesetzt.

[5] Subscriber Identity Module (SIM)

[6] Security Assertion Markup Language (SAML)

[7] Transport Layer Security (TLS) / Secure Socket Layer (SSL)

[8] Worldwide Interoperability for Microwave Access (WiMAX)

Nachdem in Kapitel 10 ein Ausblick auf die mögliche zukünftige Entwicklung gegeben ist, wird in der Schlussbemerkung in Kapitel 11 das Ergebnis dieser Arbeit zusammengefasst.

Im Anhang befinden sich eine Beschreibung der Konfiguration der Entwicklungsumgebung, Beschreibung der CA-Struktur und Anleitung zur Administration der Zertifikate, Screenshots zum ersten Prototyp und eine ausführliche Beschreibung des Aufbaus und der Konfiguration des zweiten Prototyps. Beide Prototypen inklusive des Sourcecodes sind auf einer CD beigelegt.

2 Grundlegende Technologien

In diesem Kapitel werden die Technologien beschrieben, welche im Rahmen der in dieser Arbeit zugrunde gelegten Roaming[9] Szenarien verwendet werden können. Zum einen sind dies drahtlose Netztechnologien auf denen mögliche Zugangsnetze basieren, zum anderen sind das die essentiellen Protokolle, die zum Roaming eingesetzt werden wie Mobile IP (MIP) [RFC-2977, RFC-3344, MIP-Threats] bzw. zur Erhöhung der Sicherheit der Kommunikation zwischen zwei Kommunikationspartnern wie IPSec [NaDoHa00]. Bevor die drahtlosen Netztechnologien betrachtet werden, wird zunächst der Sicherheitsbegriff definiert.

2.1 Begriffsdefinition Sicherheit

Sicherheit bedeutet in dieser Arbeit Informationssicherheit im Gegensatz zur Funktionssicherheit. Im Bereich Informationssicherheit lassen sich folgende Sicherheitsziele formulieren:

- Authentizität
- Integrität
- Vertraulichkeit
- Verbindlichkeit
- Anonymität
- Verfügbarkeit

Unter **Authentizität** einer Entität versteht man die Echtheit und Glaubwürdigkeit der Entität, welche anhand ihrer eindeutigen Identität und ihren charakterisierenden Eigenschaften überprüfbar ist. Eine sichere Authentifikation zu ermöglichen, ist der Kern der in Kapitel 4 beschriebenen Architektur. Da es sich um eine Architektur für Roaming handelt, wird in Kapitel 4 vorausgesetzt, das nur berechtigte Nutzer Zugang zu einem Endgerät haben.
Es gibt viele verschiedene Technologien, die zur Authentifikation verwendet werden. Noch immer erfreut sich das einfache Passwort großer Beliebtheit, wenn es darum geht, dass sich ein Benutzer gegenüber einem Endgerät, wie z. B. einem PC, authentifiziert. Die vierstellige Personal Identification Number (PIN) als vereinfachte Variante des Passwortes findet bei der Authentifikation gegenüber Handys oder Scheckkarten breiten Einsatz. Aber auch biometrische Verfahren werden immer öfter eingesetzt. Besonders der Fingerabdruck wird inzwischen zur Identifizierung von Nutzern gegenüber Endgeräten eingesetzt. Prinzipiell kann man zwischen Authentifizierung durch Wissen, Besitz und unveränderliche Merkmale unterscheiden. Eine Kombination von zwei dieser Arten wäre wünschenswert, wenn wirklich sichergestellt werden soll, dass nur ein bestimmter Nutzer

[9] Übergaben zwischen zwei Funkzellen eines Funkdienstes werden als horizontaler Handover bezeichnet. Wechselt man zu einem anderen Funkdienst spricht man von einem vertikalen Handover. Ein übergangsloser oder „seamless“ Handover beinhaltet zudem noch die Aussage, dass eine begonnene Sitzung auch nach dem Wechsel von einem Funknetz ins andere erhalten bleibt. Als Roaming wird eine besondere Art des Handovers bezeichnet. Roaming beinhaltet den Wechsel zwischen Netzen unterschiedlicher Betreiber.

Zugriff auf ein bestimmtes Gerät hat. Z. B. die Kombination von Handschrift und Smartcard, also unveränderlichem Merkmal und Besitz oder Fingerabdruck und Passwort, was unveränderlichem Merkmal und Wissen entspricht, wären angebracht. Bei kleineren Endgeräten, wie Handys bietet sich der Fingerabdruck als Merkmal an, während bei hinreichender Ausstattung des Endgerätes die Unterschrift wünschenswert ist. In [Haisch01, HaiSteiVie02] wird ein Ansatz vorgestellt und implementiert, der aus dem Merkmal Handschrift einen Schlüssel erzeugt. Dieser kann z. B. zum verschlüsselten ablegen von Zertifikaten bzw. den zugehörigen privaten Schlüsseln verwendet werden, welche später zur Authentifizierung eingesetzt werden. Der Vorteil eines solchen Verfahrens ist, dass der Schlüssel mit dem die Daten, welche gesichert werden, nirgendwo gespeichert werden muss. Man kann ihn auch nicht vergessen, wie ein Passwort, da er sich sozusagen immer in der Hand befindet. Der Einsatz eines derartigen Verfahrens wäre zwar zu wünschen, in der Realität wird aber nach wie vor das Passwort am häufigsten eingesetzt.

Unter **Integrität** von Daten versteht man nach [Eckert04] den Ausschluss der Möglichkeit der unbemerkten Manipulation der übertragenen Daten. Um Integrität zu gewährleisten, wird aus den relevanten Daten in der Regel eine Prüfsumme errechnet. Durch Vergleich der aus den vorliegenden Daten berechneten Prüfsumme mit einem Referenzwert lassen sich Manipulationen der Daten erkennen. Prüfsummen werden meist mittels so genannter Hashwerte gebildet, welche Daten beliebiger Länge auf einen Wert konstanter Länge eindeutig abbilden. Die am häufigsten verwendeten Verfahren sind SHA1 und MD5.

Vertraulichkeit bzw. Informationsvertraulichkeit ist nach [Eckert04] gegeben, wenn keine unautorisierte Informationsgewinnung möglich ist. Das beutet, dass Informationen nur befugten Personen zugänglich sind. Vertraulichkeit von Informationen wird im Wesentlichen durch Verschlüsselung der relevanten Daten gewährleistet. Hier werden zwei grundlegende Arten der Verschlüsselung unterschieden. Zum einen die symmetrische und zum anderen die asymmetrische Verschlüsselung. Bei den symmetrischen Verfahren wird derselbe geheime Schlüssel k für Ver- und Entschlüsselung verwendet, wie in folgender Abbildung 1 dargestellt. Dabei ist zu beachten, dass der geheime Schlüssel nur über einen sicheren Kanal ausgetauscht werden darf.

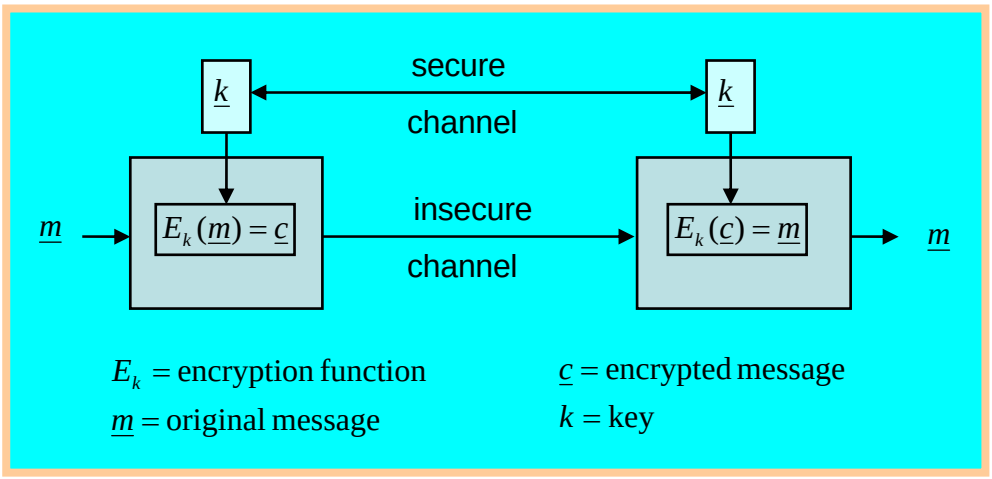


Abbildung 1: Symmetrische Verschlüsselung

Bei den asymmetrischen Verfahren wird zur Verschlüsselung ein öffentlich zugänglicher Schlüssel der so genannte „Public Key“ eingesetzt und zur Entschlüsselung ein privater Schlüssel „Private Key“ genannt, der nur den Berechtigten zugänglich sein darf. Folgende Abbildung 2 zeigt das Prinzip der asymmetrischen Verschlüsselung mit einem öffentlichen Schlüssel *e* und einem privaten Schlüssel *d*.

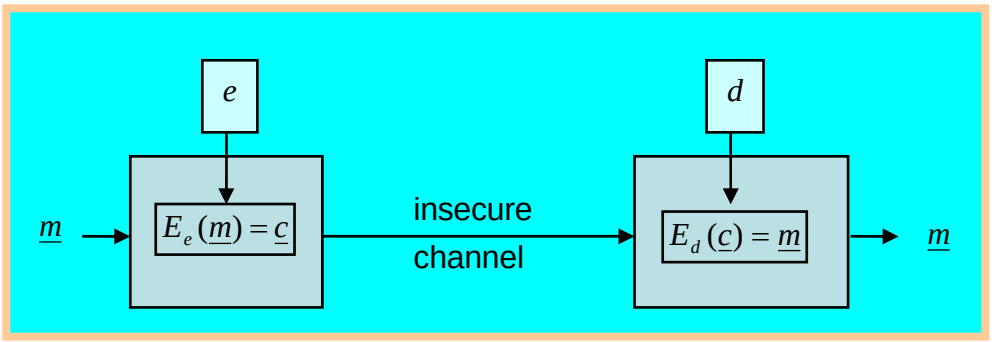


Abbildung 2: Asymmetrische Verschlüsselung

Die unterschiedlichen Funktechnologien verwenden verschiedene Verschlüsselungsverfahren. Wie man unterschiedliche gewünschte und der Leistung des Endgerätes angepasste Sicherheitsstufen erreichen kann, wird in Abschnitt 2.3.2.4 beschrieben.

Verbindlichkeit impliziert, dass Entitäten die von ihnen durchgeführten Aktionen nicht abstreiten können. Sie wird i. d. R. mit digitalen Signaturen erreicht. Hierbei kommt meist eine asymmetrische Kryptographie zum Einsatz. Die relevanten Daten oder eine zugehörige Prüfsumme werden mit einem, nur dem Signierer zugänglichen privaten

Schlüssel verschlüsselt. Mit einem öffentlichen Schlüssel kann überprüft werden von wem die Daten stammen. Hierbei werden Public Key Infrastrukturen eingesetzt, mit Hilfe derer etwaige Nutzer die Zugehörigkeit und Gültigkeit von Schlüsseln von einer vertrauenswürdigen neutralen Instanz bestätigt bekommen. Die Schlüsselverwaltung fällt unter anderem in den Bereich des Trust-Managements. Idealerweise sollte ein mobiles Endgerät hierfür mit einem Trusted Platform Module (TPM) ausgestattet sein, welches zum ablegen privater bzw. geheimer Schlüssel dienen kann [www.trustedcomputinggroup.org]. Bei der in dieser Arbeit entwickelte Roaming Architektur wird Nicht-Abstreitbarkeit oder Verbindlichkeit berücksichtigt. Dies ist in Kapitel 4 näher erläutert.

Anonymisierung ist das Verändern personenbezogener Daten derart, dass eine Zuordnung der persönlichen Daten zu Personen mit vertretbarem Aufwand nicht möglich ist [Eckert04]. **Anonymität** steht im Gegensatz zur Authentizität und ist zusammen mit nicht Abstreitbarkeit gleichzeitig nicht möglich. Eine abgeschwächte Form der Anonymisierung ist die Pseudonymisierung. Hierbei werden personenbezogene Daten durch eine Zuordnungsvorschrift so verändert, dass die Daten ohne Kenntnis der Zuordnungsvorschrift nicht mehr einer natürlichen Person zugeordnet werden können. Analog zu den anderen Schutzzielen gibt es keine expliziten Mechanismen, wie eine Verschlüsselung, die Anonymität gewährleisten. Die unten beschriebenen Technologien gewährleisten an sich keine Anonymität. Da es nicht Ziel dieser Arbeit ist, eine Roaming Architektur zu entwickeln, die Anonymität bietet, wird auf die Anonymität im Folgenden nicht weiter eingegangen.

Verfügbarkeit gewährleistet ein System gemäß [Eckert04], wenn Subjekte, die authentifiziert und autorisiert sind, nicht unautorisiert an der Wahrnehmung ihrer Berechtigungen gehindert werden können. Denial of Service (DoS) bzw. verteilte DoS (DDoS) Angriffe, die die Verfügbarkeit mindestens einschränken können bzw. das Zugreifen auf bestimmte Ressourcen völlig unterbinden können, sind bei drahtloser Kommunikation besonders einfach. Bei auf Lichtwellen basierender Kommunikation genügt es den Lichtstrahl zu unterbrechen und bei Funk kann die Kommunikation mit einem hinreichend starken Störsender unterbunden werden. Einen Mechanismus aus den Welten der Technik oder der Informatik, der dies verhindern kann, gibt es nicht. Die einzige Möglichkeit eine derartige Störung zu unterbinden ist es, den Störsender bzw. Unterbrecher zu lokalisieren und zu deaktivieren bzw. zu entfernen. Die unten aufgeführten Funk-Technologien können prinzipiell alle im Rahmen von Zugangsnetzen eingesetzt werden. Wie unten dargestellt senden alle im Industrial, Scientific and Medical (ISM) Frequenzband, einem Frequenzbereich, der nicht staatlich reguliert ist und lizenzfrei genutzt werden darf, wenn Auflagen was die Sendeleistung und die Störung benachbarter Frequenzen angeht erfüllt werden. Daher gibt es, was die unten beschriebenen Funktechnologien angeht, keine derart signifikanten Unterschiede hinsichtlich der Verfügbarkeit, die den Ausschluss einzelner Technologien rechtfertigen würde.

Zunächst werden die drahtlosen Netztechnologien dargestellt.

2.2 Drahtlose Netztechnologien

Die wesentlichen drahtlosen Übertragungstechnologien, die innerhalb von Roaming Szenarien denkbar sind, basieren auf Funkwellen. Alternativ dazu sind auch Lichtwellen zur kabellosen Datenübertragung einsetzbar. Insbesondere die Datenübertragung über infrarotes Licht hat sich dank des Infrared Data Association kurz IrDA– Standards durchgesetzt. Allerdings eignet die Infrarot-Technologie sich schlecht für mobile Geräte, welche von umherwandernden Personen verwendet werden, als Netzzugangstechnologie. IrDA wird i. d. R. eingesetzt, wenn eine kabellose Datenübertragung zwischen zumindest temporär für die Kommunikationsdauer fest stehenden Entitäten gewünscht ist. Dies und die Tatsache, dass IrDA immer mehr durch die weiter unten beschriebenen Bluetooth Technologie verdrängt wird, ist der Grund dafür, dass auf IrDA hier nicht weiter eingegangen wird.

Die wesentlichen Charakteristiken der unterschiedlichen Funktechnologien nach denen man sie unterscheidet sind zunächst einmal ihre Reichweite und die Übertragungsgeschwindigkeiten. Weitere Eigenschaften sind der Leistungsbedarf und die maximale Geschwindigkeit mit der sich ein Nutzer mit mobilem Gerät bewegen darf. Wenn man die Reichweite als Grundlage nimmt, kann man Funknetze grob in „wireless personal area networks“ kurz WPANs, „wireless local area networks“ kurz WLANs, „wireless metropolitan area networks“ kurz WMANs und „wireless wide area networks“ kurz WWANs unterteilen. Zu den WWANs zählen das in Abschnitt 2.2.5 beschriebene IS-95, das in Abschnitt 2.2.7 beschriebene GSM, das in Abschnitt 2.2.8 beschriebene GPRS und das in Abschnitt 2.2.9 beschriebene UMTS. Die WPANs sind mit Bluetooth in Abschnitt 2.2.6 und mit DECT in Abschnitt 2.2.4 vertreten. Abschnitt 2.2.1 beinhaltet die unterschiedlichen Standards für WLAN. Das in Abschnitt 2.2.2 beschriebene WiMAX zählt zu den WMANs. Man kann WiMAX im Prinzip als WLAN mit einer vergleichsweise höheren Reichweite ansehen.

2.2.1 WLAN

Ein Wireless Local Area Network kurz WLAN ist wie der Name schon sagt zunächst einmal ein drahtloses Local Area Network kurz LAN [Walke01b]. Das Institute of Electrical and Electronics Engineers kurz IEEE hat hierzu eine ganze Reihe von Standards erarbeitet. Die Arbeitsgruppe, die sich mit WLANs beschäftigt ist die Gruppe 802.11. Die Tabelle 1 gibt einen Überblick über die in diesem Zusammenhang relevanten Standards des IEEE.

Tabelle 1: IEEE Standards für WLAN, WPAN, WMAN,MBWA u.a.

IEEE Standard	Art	Übertragungsrate	Frequenzbereich	Beschreibung
802.11	WLAN	2 MBit/s	2,4 GHz	Protokoll und Übertragungsverfahren für drahtlose Netze von 1997
802.11a		54 MBit/s	5 GHz	12 nicht-überlappende Kanäle, Modulation: Orthogonal Frequency Division Multiplexing (OFDM)
802.11b		11 MBit/s	2,4 GHz	3 nicht-überlappende Kanäle
802.11c				Wireless Bridging zwischen AccessPoints
802.11d				World Model
802.11e				„MAC Enhancements" Erweitert WLAN um Quality of Service (QoS) - Priorisierung von Datenpaketen, z.B. für multimediale Anwendungen, Streaming und QoS zur Sprachübertragung im Funknetz
802.11f				Roaming zwischen AccessPoints verschiedener Hersteller
802.11g		54 MBit/s	2,4 GHz	Im Vergleich zu 11b höhere Übertragungsrate, Modulation OFDM
802.11h		54 MBit/s	5 GHz	Ergänzungen zu 802.11a für Europa: Spektrum Management mit DFS und TPC, Modulation OFDM
802.11i				Sicherheitsmechanismen, als "WPA2" von der WiFi Alliance übernommen.
802.11j			4.9GHz-5GHz	japanische Variante von 802.11a
802.11k				Bessere Messung, Auswertung und Verwaltung der Funkparameter (z.B. Signalstärke), soll z.B. Ortsbezogene Dienste (location-based services) ermöglichen
802.11m				Zusammenfassung früherer Ergänzungen, Bereinigung von Fehlern aus vorausgegangenen Spezifikationen (Maintenance)
802.11n		>100MBit/s		geplante Erweiterung für ein zukünftiges, schnelleres WLAN mit 108Mbit/s - 320MBit/s (600MBit/s TGn Sync) Modulation OFDM
802.15	WPAN	>480MBit/s	0,1 - 10.6GHz	Bluetooth Nachfolger für kurze Distanzen bis ca. 12 Meter, Wireless USB, Personal Area Network

IEEE Standard	Art	Übertragungsrate	Frequenz-bereich	Beschreibung
802.15.3		480MBit/s	2,4 GHz	"Garantierter Level of Service" Streamen von Daten über WLAN Keine Interferenzen zu anderen Funktechniken, nachdem eine Verbindung zwischen Client und Server aufgebaut ist. IEEE 1394 über IEEE 802.15.3 , drahtlose FireWire Protokolle
802.16	WMAN	Bis 70MBit/s	10-60 GHz	"Wireless MANs" Air Interface for Fixed Broadband Wireless Access Systems. WiMax Erste Produkte werden erwartet ab Ende 2005
802.16a		Bis 134MBit/s	2-11 GHz	Breitband Anwendungen, WiMAX für unbewegliche Empfangseinheiten im Frequenzbereich <=11 GHz
802.16b			5-6 GHz	Licensed Exempt Frequencies, Diese Gruppe läuft auch unter der Bezeichnung WirelessHUMAN (High Speed Unlicensed MAN)
802.16c			10-66GHz	10-66 GHz Profile, Konformitätsstandards für 802.16 um die Interoperabilitäts-Spezifikationen zu erleichtern
802.16d		Bis 70MBit/s		der um die im WiMAX-Forum erarbeiteten Ergänzungen bereicherte Standard wird seit 2004 "IEEE 802.16-2004" genannt
802.16e				„Mobile WirelessMAN" Standards für mobile Nutzung. Erweiterungen zum 802.16a PHY/MAC, um mobile Operationen zu ermöglichen. Draft 2004. Standard ab Mitte 2005. Beinhaltet bewegliche Empfangseinheiten mit Geschwindigkeiten von bis zu 120 Km/h, d.h. Mobilität
802.20	MBWA	> 1 Mbit/s		Mobile Broadband Wireless Access (MBWA): In Autos und Zügen mit bis 250 km/h - hohe Reichweite. Datenraten von > 1 Mbit/s je Benutzer; Der Zeitplan sieht eine Verabschiedung dieser Norm bis Ende 2005 vor.

Fortsetzung von Tabelle 1

2.2.1.1 IEEE Standards für WLAN

Im Folgenden werden zunächst die IEEE Standards zu WLAN erläutert. Zusätzlich werden zum Vergleich noch die Standards IEEE 802.15, IEEE 802.16 und IEEE 802.20 erklärt. Der Standardisierungsprozess ist für diese drei Standards noch nicht endgültig abgeschlossen. Trotzdem sollen diese hier nicht unerwähnt bleiben, um einen Vergleich zu WLAN zu ermöglichen.

2.2.1.1.1 IEEE 802.11a

Geräte nach dem Standard 802.11a aus dem Jahr 1999 arbeitet im 5 GHz Bereich. Durch den Frequenzbereich von 5,725 GHz bis 5,850 GHz sind sie relativ störungsfrei. Auf Grund der Tatsache, dass in dem Frequenzbereich auch Netze des Militärs und zur Flugsicherung senden sind in Europa dem Standard 802.11a konforme Geräte nur für den Einsatz innerhalb von Gebäuden und mit einer gedrosselten Sendeleistung zugelassen. Die Reichweite ist sehr gering und liegt zwischen 15 und 25 Metern. Die maximale Übertragungsrate beträgt 54 MBit pro Sekunde.

2.2.1.1.2 IEEE 802.11b

Ebenfalls ein Wireless LAN Standard von 1999, welcher im 2,4 GHz Bereich angesiedelt ist. Trotz der im Vergleich zu IEEE 802.11a geringen Übertragungsrate von 11 MBit pro Sekunde ist dieser WLAN-Standard wesentlich verbreiteter und findet sich an vielen Universitäten und auch bei öffentlichen WLAN Hot-Spots wieder. Die Vorteile sind unter anderem die höhere Reichweite von bis zu 300m, die mit externer Antenne im Outdoor-Einsatz erreicht werden kann, und auch die Kompatibilität zum IEEE 802.11g Standard. Ein wesentlicher Nachteil von IEEE 802.11b ist jedoch das Frequenzband. Da bei 2,4 GHz auch andere Geräte arbeiten und unter anderem auch Bluetooth dort angesiedet ist, kann es zu Störungen kommen.

2.2.1.1.3 IEEE 802.11c

IEEE 802.11c ist ein Standard für die drahtlose Koppelung unterschiedlicher Netzwerk-Topologien. IEEE 802.11c wurde entwickelt um mehrere Netzwerke mittels Wireless LAN verbinden zu können. Die Mac-Adresse dient hierbei als Grundlage der Identifikation der Gegenstelle.

2.2.1.1.4 IEEE 802.11d

Der IEEE 802.11d Standard wird auch gerne als "World Mode" bezeichnet. Er regelt die regionalen technischen Unterschiede. Hierzu gehören u. a. die Anzahl und die Auswahl der für die Nutzung im entsprechenden Land freigegebenen Kanäle. Ebenfalls geregelt wird die Auswahl der Basistechnologie, d.h. ob IEEE 802.11 a, b, g oder h verwendet werden darf. Ein 802.11d kompatibles Endgerät arbeitet nachdem der Endnutzer seinen aktuellen Standort über eine Länder bzw. Regionalauswahl spezifiziert hat automatisch mit den jeweils zugelassenen Standards.

2.2.1.1.5 IEEE 802.11e

Der IEEE 802.11e Standard sieht Neuerungen für IEEE 802.11 a, h und g vor und erweitert diese unter anderem um QOS (Quality Of Service). Mit den Änderungen sollen die WLAN-Standards besser auf die Nutzung von Multimedia und Voice over IP (VOIP) abgestimmt werden und in der Lage sein eine gewisse Datenrate zu garantieren sowie minimale Schwankungen bei der Paketlaufzeit. QOS erlaubt es z. B. die Datenpakete für Internet-Telefonie bevorzugt zu versenden und dadurch geringere Verzögerungen zu haben.

2.2.1.1.6 IEEE 802.11f

Der IEEE 802.11f Standard sieht Verfahren für das Roaming nach dem Inter Access Point Protocol (IAPP) von Clients zwischen verschiedenen Accesspoints vor. Mittels 802.11f wird es möglich innerhalb eines großen drahtlosen Netzwerkes seinen Standort über die Reichweite eines einzelnen Accesspoints hinaus zu verändern. Roaming bedeutet hier, dass die Netzwerk-Verbindung ohne Abbruch von einem Accesspoint auf den anderen übergeht.

2.2.1.1.7 IEEE 802.11g

Der 802.11g Standard von 2002/2003 ist vollkommen abwärtskompatibel zu dem älteren 802.11b Standard und arbeitet ebenfalls auf Frequenzen von 2,4 GHz bis 2,4835 GHz im 2,4 GHz Frequenzband. Die Geschwindigkeit ist wie bei 802.11a auf maximale 54 MBit pro Sekunde beschränkt. Die Sendeleistung und dementsprechend die Reichweite entspricht der des 802.11b Standards. Dank der Kompatibilität lassen sich 802.11g Accesspoints (APs) und Router problemlos in ein bestehendes 802.11b-Netz integrieren. Wenn eine 802.11b konforme Komponente wie z.B. ein AP und eine 802.11g konforme Komponenten wie z.B. eine WLAN-Karte für einen Laptop aufeinander treffen, arbeiten sie problemlos zusammen. Sie einigen sich automatisch auf einen Betriebsmodus konform zu IEEE 802.11b.

2.2.1.1.8 IEEE 802.11h

Der 802.11h Standard ergänzt den 802.11a Standard um Dynamic Frequency Selection (DFS) und Transmit Power Control (TPC) und erlaubt nach der RegTP-Nutzungsverordnung für Frequenzen eine maximale Sendeleistung von bis zu 200 mW. Da 802.11h jedoch weiterhin den gleichen Frequenzbereich wie 802.11a nutzt ist auch bei Verwendung von dem Standard 802.11h konformen Geräten nur der Indoor-Betrieb zugelassen.

2.2.1.1.9 IEEE 802.11i

Bei IEEE 802.11i geht es um Authentifizierung und Verschlüsselung für die IEEE 802.11 a/b/g/h Standards. Mittels IEEE 802.11i wird versucht die Sicherheit von WLANs zu erhöhen. IEEE 802.11i sieht unter anderem die Authentifizierung nach IEEE 802.1x (Extensive Authentication Protocol) vor und auch die Verschlüsselung nach AES (Advanced Encryption Standard). 802.11i ist abwärts kompatibel zu 802.11 und wird oft als WiFi bezeichnet.

2.2.1.1.10 IEEE 802.11j

Bei IEEE 802.11j handelt es sich um die siebte Ergänzung des IEEE 802.11 Standards. Es geht hier um die Anpassung von 802.11 für den Einsatz in Japan – genauer gesagt die Nutzung der 4,9 GHz und 5 GHz Frequenzbänder für indoor-, outdoor- und Mobilkommunikation.

2.2.1.1.11 IEEE 802.11k

Der Standard IEEE 802.11k ermöglicht bessere Messung, Auswertung und Verwaltung der Funkparameter, wie z.B. Signalstärke. Dies soll z.B. ortsbezogene Dienste sogenannte „location-based services“ ermöglichen.

2.2.1.1.12 IEEE 802.11m

IEEE 802.11m besteht aus der Zusammenfassung früherer Ergänzungen. Es werden außerdem Fehler aus vorausgegangenen Spezifikationen berichtigt.

2.2.1.1.13 IEEE 802.11n

IEEE 802.11n beinhaltet geplante Erweiterung für ein zukünftiges, schnelleres WLAN mit einer Übertragungsrate von 108Mbit/s bis 320MBit/s. Zur Modulation ist OFDM vorgesehen.

2.2.1.1.14 IEEE 802.15 und IEEE 802.15.3

Der Standard IEEE 802.15 definiert Technologien für Wireless Personal Area Networks (WPANs). Im Vergleich zu WLAN handelt es sich um eine neue Technologie mit geringerer Reichweite aber mit höheren Datenraten als sie die aktuelle WLANs Die Technik ist so ausgelegt, dass sie nicht mit Signalen nach IEEE 802.11, 802.16 und 802.20 oder anderen Funkdiensten sollte.
Intel hat auf dem Intel Developer Forum seine Pläne für ein drahtloses Hochgeschwindigkeitsnetz auf Basis des IEEE 802.15 Standards vorgestellt. "Ultra-Wideband" (UWB) soll PC-Peripherie, Unterhaltungselektronik und mobile Geräte mit Datentransferraten von bis zu 480 Megabit/s verbinden. Es eignet sich laut Intel z.B. für die Übertragung von Multimediainhalten wie Videos vom digitalen Videorekorder an den Fernseher. Die Technologie verwendet ein breites Band des Funkfrequenzspektrums für die Übertragung von Daten innerhalb eines kleinen Umkreises, wie etwa im Büro oder zu Hause. Interferenzen mit anderen kabellosen Systemen wie WLAN, WiMAX, oder Mobiltelefonen soll eine gepulste Datenübertragung vermeiden. Dabei benötige UWB nur wenig Energie. Die Bluetooth Special Interest Group, kurz SIG, will mit den Entwicklern des möglichen neuen Funkstandards UWB mit dem Ziel der Kombination aus UWB und Bluetooth zusammenarbeiten. Es könnte auf den unteren Netz-Schichten die schnellere UWB-Technik verwendet werden, während auf den oberen Schichten die bekannten Bluetooth-Profile zum Einsatz kommen, welche z.B. die Kommunikation zwischen Endgeräten wie Handy und Headset definieren.
Basierend auf der „Ultra Wideband“ kurz UWB Technologie analog zum IEEE 802.15.3 Standard wird Wireless USB [WUSB05a, WUSB05b, WUSB06] - kurz WUSB - als eine neue drahtlose Erweiterung zum USB-Standard entwickelt, die im 3,1 GHz bis 10,6 GHz Frequenzbereich arbeitet. Am 12. Mai 2005 wurde die erste Spezifikation für WUSB vorgestellt [WUSB05a]. WUSB soll die Übertragungssicherheit und die Geschwindigkeit von drahtgebundenen Verbindungen mit der einfachen Nutzung drahtloser Technologien kombinieren. WUSB könnte ein Nachfolger für Bluetooth werden. WUSB soll bei Abstand von 3 Metern zwischen Sender und Empfänger analog zu USB 2.0 480 MBit/s

übertragen; Bei einer Entfernung von 10 Metern kann man noch eine Übertragungsrate von 100 MBit/s erreichen.

Die in diesem Abschnitt genannten Standards befinden sich noch in der Entwicklungsphase, was dazu führt, dass sie in dieser Arbeit nicht weiter betrachtet werden können. Soweit möglich, werden im nächsten Abschnitt die Sicherheitsmechanismen von „certified wireless USB“ (CWUSB) nach [WUSB05a, WUSB05b] dargestellt.

2.2.1.1.14.1 Sicherheit von CWUSB

Die USB Kern-Spezifikation bietet im Moment noch keine Sicherheitsmechanismen [WUSB05a]. Anwendungen sollten daher wenn notwendig oberhalb von USB Sicherheitsmechanismen implementieren. Bei der Definition von CWUSB [WUSB05a, WUSB05b] wird aber eine Architektur für grundlegende Sicherheit dargestellt. Die drahtlose USB Verbindung soll eine einem Kabel entsprechende Sicherheit bieten. Bei der erstmaligen Verbindung eines Endgerätes mit einem Host soll hierfür eine Authentifizierung durchgeführt werden. Je nach Möglichkeiten des Endgerätes werden verschiedene Authentifizierungsmöglichkeiten bereitgestellt. Hierbei werden drei Fälle unterschieden:

1. Das Endgerät hat einen sicheren Kanal für die Schlüsselverteilung. Beispiele für einen solchen Kanal wären ein festes USB-Kabel, eine Speicherkarte oder irgendeine Nutzerschnittstelle über die ein Nutzer adäquate Daten eingeben kann. Ein Endgerät, welches einen solchen Kanal für den Schlüsselaustausch zu Verfügung stellt, muss nur einmal authentifiziert werden. Dabei wird dann ein so genannter Verbindungskontext zum Endgerät übertragen. Dies entspricht in etwa der Kopplung bei Bluetooth. Unabhängig davon welche Art sicherer Kanal verwendet wird, ist auf jeden Fall eine Nutzerinteraktion erforderlich.
2. Das Endgerät verfügt über einen fest eingebauten symmetrischen Schlüssel, der dem Endgerät eindeutig zugeordnet werden kann. Dieser ist normalerweise bei der Fertigung in die Hardware fest integriert.
3. Das Endgerät beinhaltet ein ihm eindeutig zugeordnetes Schlüsselpaar bestehend aus einem privatem und einem öffentlichen Schlüssel. Dies wäre z. B. bei einem Gerät mit eingebautem TPM der Fall.

In [WUSB05a] werden für die Authentifikation der drei unterschiedlichen Gerätetypen so genannte Zeremonien vorgeschlagen. Zeremonien entsprechen Netz-Protokollen mit dem Unterschied, dass in Zeremonien im Gegensatz zu Protokollen immer auch Nutzer involviert sind. Die Zeremonien für die drei oben genannten Gerätetypen sind in Abbildung 3, Abbildung 4 und Abbildung 5 dargestellt und in [WUSB05a] detailliert beschrieben. Nach erfolgter Authentifikation kommunizieren der Host und das Endgerät über eine AES-128 CCM verschlüsselte Verbindung.
Auf diese Weise wird zwar eine zu einem Kabel adäquate Sicherheit erreicht, aber auf Grund der Tatsache, dass immer der Nutzer involviert ist, eignet sich die WUSB Sicherheit nicht für ein übergangsloses Roaming. Durch die Nutzerinteraktion wird die Übergangslosigkeit unmöglich gemacht. Wenn man in eine Roaming Architektur ein auf

WUSB basierendes Zugangsnetz integrieren möchte, muss man daher wenn übergangsloses Roaming erreicht werden soll auf die in der WUSB Spezifikation vorgeschlagenen Sicherheitsmechanismen zu Authentifikation verzichten und die WUSB Schnittstelle als unsicheren Kanal nutzen, über den dann auf höheren Ebenen entsprechende Sicherheitsmechanismen implementiert werden.

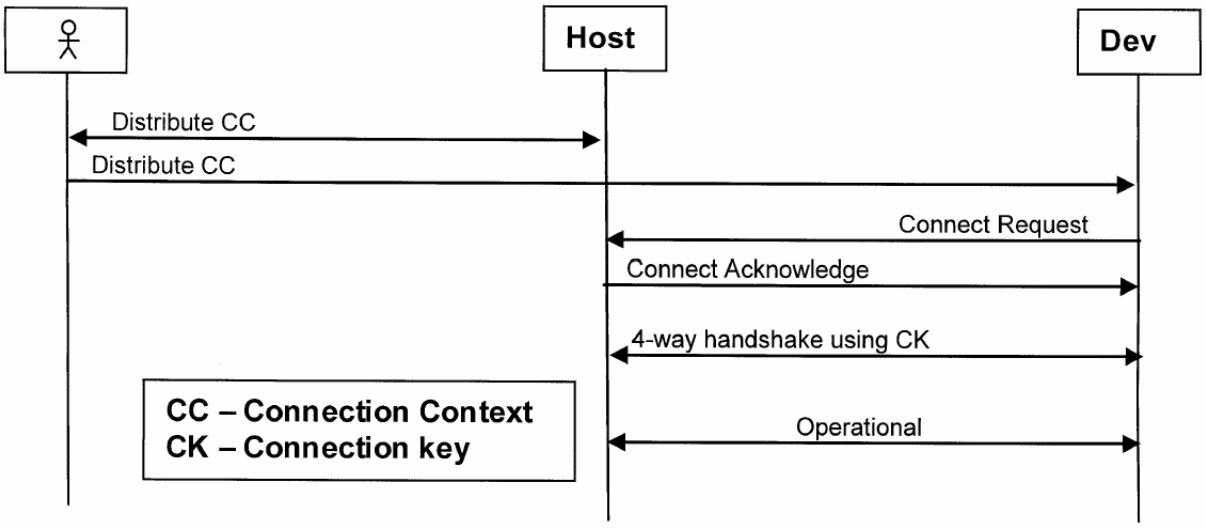


Abbildung 3 Zeremonie für Endgeräte mit sicherem Kanal [WUSB05a]

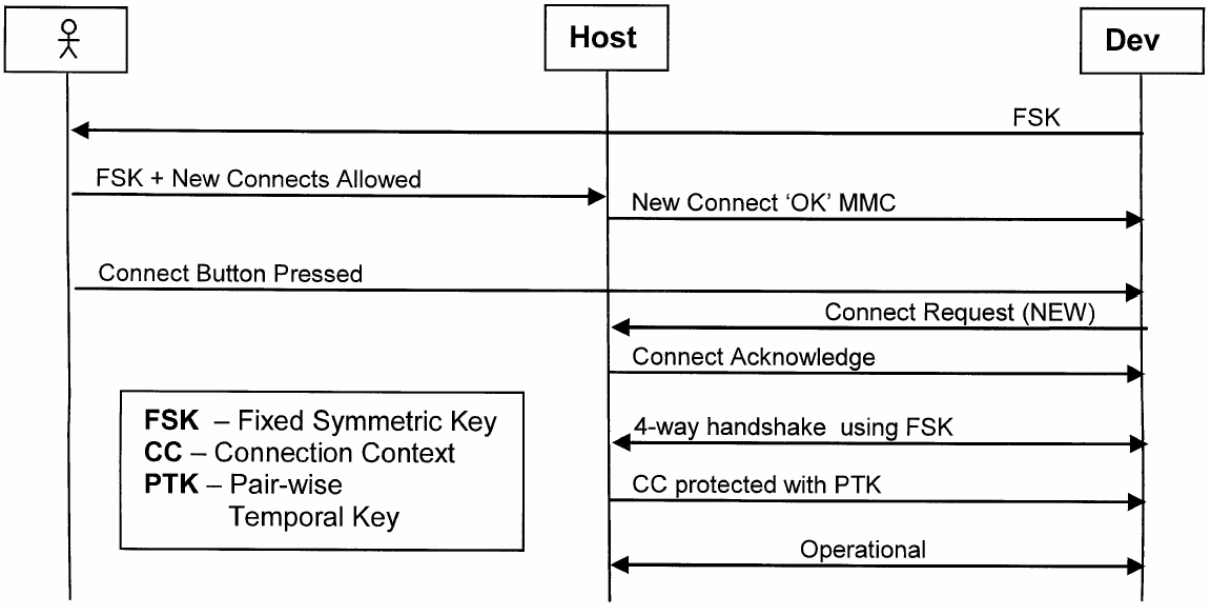


Abbildung 4 Zeremonie für Endgeräte mit festem symmetrischem Schlüssel [WUSB05a]

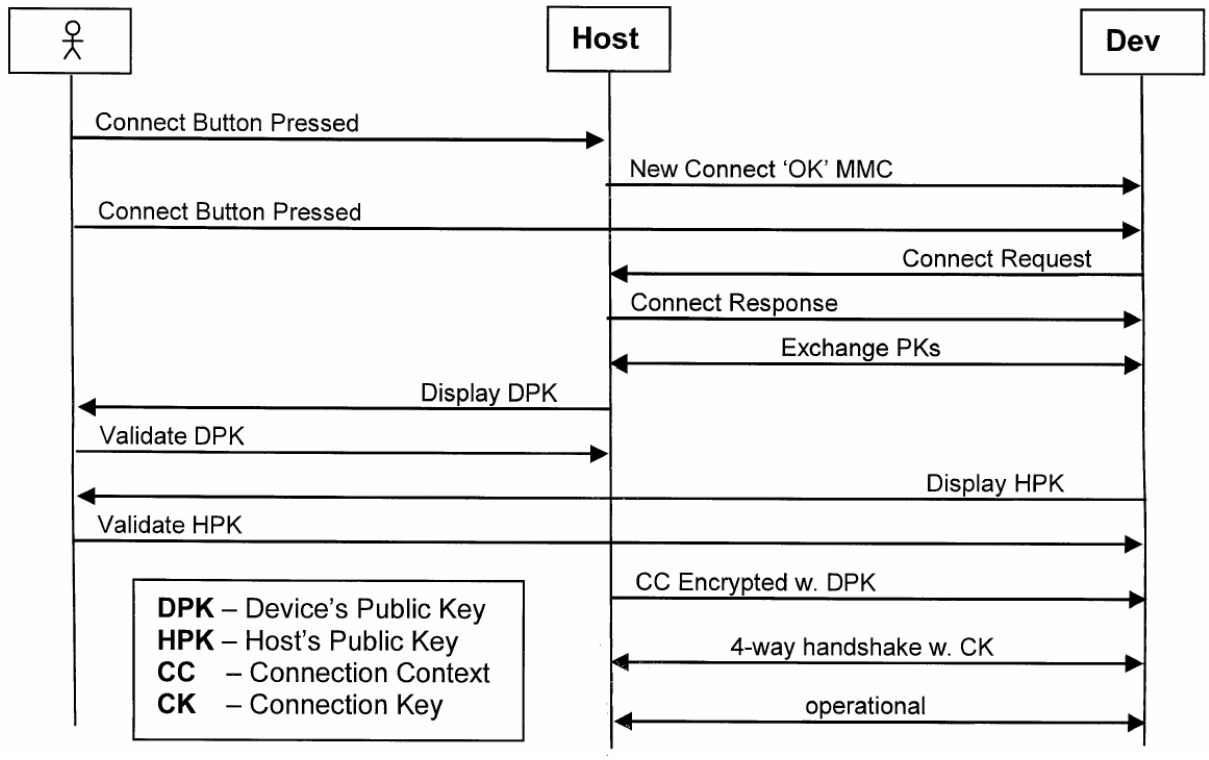


Abbildung 5 Zeremonie für Endgeräte mit Schlüsselpaar [WUSB05a]

2.2.1.1.15 IEEE 802.16a-d

Der IEEE 802.16 Standard wurde am 8. April 2004 veröffentlicht. Er definiert die Spezifikationen für die Luftschnittstelle von drahtlosen Metropolitan Area Networks so genannten Wireless MANs. Er beschreibt also keine WLANs. Im Abschnitt 2.2.2 wird der IEEE 802.16 Standard genauer erläutert. Die Tabelle 2 zeigt schließlich die wesentlichen Unterscheide von WMAN zu WLAN.

Tabelle 2: Unterschied von WLAN zu WiMAX

WLAN (IEEE 802.11)	WMAN (802.16)
Für lokale Netze gedacht	Als „letzte Meile“ vorgesehen
Reichweite 200-300m	Max. Reichweite 50 km
Für den Einsatz innerhalb von Gebäuden gedacht	Für den Einsatz außerhalb von Gebäuden gedacht
Geringe Anzahl von Teilnehmern (<100)	Große Anzahl von Teilnehmern (>100)
Datenrate bis 54Mbit/s	Datenrate bis 100Mbit/s
QoS Unterstützung durch 802.11h	QoS in MAC enthalten
Nur Nutzung lizenzfreier Frequenzbänder	Auch Nutzung lizenzpflichtiger Frequenzbänder

In Deutschland wurde von der RegTP im 3,5-GHz-Bereich für WiMAX zwei Frequenz-Blöcke zwischen 3410 bis 3452 MHz und 3510 bis 3552 MHz frei gegeben.

Im Rahmen der Versteigerung der UMTS-Lizenzen zur Nutzung eines reservierten Frequenzspektrums wurden auch Lizenzen zur Nutzung für Punkt-zu-Punkt-Richtfunk – kurz PmP-RiFu namens Wireless Local Loop kurz WLL versteigert. Über 50 Firmen nahmen daran teil und hegten die Absicht eine Infrastruktur auf Richtfunkbasis für einen schnellen Internet-Zugang sowie Standortvernetzung aufzubauen. Da die Geschäftsmodelle sich als nicht wirtschaftlich heraus stellten, kam kein derartiges Angebot bzw. Produkt bisher zustande. Zwischenzeitlich wurden technische Fortschritte erzielt und mit WiMAX eine flexiblere Funktechnik entwickelt, die auch als Standard festgelegt ist. Bisher ist allerdings noch nicht klar, wie die Frequenzen in Deutschland vergeben werden.

In der Diskussion stehen drei Verfahren:

- Allgemeinzuteilung und der Betrieb eines WiMAX-Access-Points mit Zwangsregistrierung für jedermann.
- Einzelzuteilung mit Vergabe einer Betriebserlaubnis für eine bestimmte Region.
- Vereinfachte Zuteilung, bei der sich die Antragsteller untereinander einigen müssen.

Die Zukunftsaussichten von WiMAX scheinen sich folgendermaßen zu entwickeln:

Die Anzahl installierter Systeme wächst weltweit. Allerdings handelt es sich um Pilotversuche und Testprojekte mit proprietären Systemen, die mit dem WiMAX-Standard nicht kompatibel sind. Bisher ist noch keine Hardware zertifiziert. Erst seit Julie 2005 gibt es dafür ein Labor. Bis Ende 2005 sollen die ersten standardkonformen Geräte auf den Markt kommen.
Ob WiMAX ein mit einheitlicher Technik ein weltweiter Erfolg wird, ist stark von den staatlichen Regulierungsbehörenden und deren Frequenzvergabe abhängig. Die Frequenzzuteilung in den USA, Europa und Asien ist weitgehendst uneinheitlich. Zu viele Frequenzbänder möchten die Hersteller von WiMAX-Geräten vermeiden. Multiband-Geräte sind aufwendiger in der Entwicklung und Herstellung. Das macht die Geräte teurer, drückt dadurch die Verkaufszahlen und macht WiMAX weder für Hersteller noch für den Handel interessant.
Fraglich ist das Interesse der Anwender. WiMAX ist auf lizenzpflichtige Frequenzen angewiesen, die von kommerziell ausgerichteten Netzbetreibern verwendet werden. Diese bieten einen Zugang nur gegen bares Geld an. Ein offenes System, wie es von WLAN bekannt ist, wo jeder einen Access Point oder Hot-Spot betreiben kann, ist eher unwahrscheinlich. Somit reduziert sich der Nutzen für den Anwender, sich ein mit WiMAX integriertes Notebook anzuschaffen. Das bedeutet eine geringe Marktdurchdringung und wenig Kunden für die Netzbetreiber. Mobilfunk-ähnliche

Angebote mit subventionierten Datenkarten sind die Folge, was die Nutzung von WiMAX auf einen DSL-Ersatz reduziert.

Die Planung wurde in einigen Pilotstädten eingestellt, da dort nun stattdessen ein T-DSL-Ausbau stattfinden wird.

In Deutschland wird die WiMAX-Technik in der zweiten Jahreshälfte 2005 innerhalb eines Pilotbetriebes in Nordrhein-Westfalen von der Deutschen Telekom eingeführt. Teile der Stadt Sankt Augustin bei Bonn werden mit WiMAX versorgt werden.
Im westfälischen Selm entsteht das erste kommerzielle WiMAX-Netz in Deutschland. Der Backbone besteht dabei allerdings aus einem Pre-WiMAX-Standard; die Anbindung der Endnutzer erfolgt durch das bekannte WLAN. Im sauerländischen Finnentrop das ist das erste kommerzielle WiMAX (pre-release) Netzwerk, welches bis zum Endgerät reicht mit einigen Funklöchern in Betrieb genommen worden. Im Gegensatz zu Selm wird dort kein WLAN verwendet. Dadurch besteht für den Internet-Nutzer neben der höheren Datensicherheit auch der Vorteil, Quality of Service (QoS) für Sprachübertragungen (VoIP) geboten zu bekommen. Der Anbieter Airtraxx hat am 11.8.2005 die ersten Kunden für sein WiMAX Netz in Finnetrop freigeschaltet. In der ersten Ausbaustufe können angeblich bis zu 750 Kunden über zwei Basisstationen bedient werden. Ein weiterer Ausbau auf die Nachbargemeinden Lennestadt und Attendorn soll bereits in der Planung sein. Auch in den beiden Gemeinden Seevetal und Rosengarten (Niedersachsen, südlich von Hamburg) wird ein WiMAX-Netz aufgebaut werden, welches direkt zum Kunden geht, ohne eine WLAN-Basisstation zum Weiterverteilen.
Unter dem Namen AirMAX wird von Arcor mit Kaiserslautern zum ersten Mal in Deutschland eine komplette Großstadt von einem der führenden Telekommunikationsanbieter mit einem WiMAX-Netz versorgt. AirMAX ist etwas teurer als der vergleichbare DSL-Anschluss und wird bereits 2005 angeboten.

Im August 2005 wurde WiMAX unter den Namen MAXXtelecom durch die DBD in Heidelberg flächendeckend in Betrieb genommen. Mit der Fachhochschule Heidelberg habe sich ein erster Kunde für das neue Angebot entschieden. Die Standard-Produkte haben eine symmetrische Bandbreite von 1,5 MBit/s bis zu 3,5 MBit/s. Darüber hinaus bietet DBD individuelle Produkte mit Bandbreiten von bis zu 155 MBit/s an. Eine Einführung von Voice over IP ist in Planung.

2.2.1.1.16 IEEE 802.20

Die Arbeitsgruppe 802.20 des IEEE beschäftigt sich mit mobilen, drahtlosen Breitbandzugängen. Diese Gruppe führt die Bezeichnung Mobile Broadband Wireless Access (MBWA). Die Abbildung 6 zeigt die wesentlichen Entwicklungsziele für MBWA:

Eigenschaften MBWA/802.20	Übertragungsband 1,25 MHz	5 MHz
Datenrate		
Downlink	> 1 Mbit/s	> 4 Mbit/s
Uplink	> 0,3 Mbit/s	> 1,2 Mbit/s
Aggregierte Datenrate je Zelle		
Downlink	> 4 Mbit/s	> 16 Mbit/s
Uplink	> 0,8 Mbit/s	> 3,2 Mbit/s
Frequenzbereich	< 3,5 GHz	
Dauerspektraleff.	> 1 bit/s/Hz/Zelle	
Mobilitätsänderung	> 250 km/h	

Abbildung 6: Entwicklungsziele für MBWA

Der IEEE 802.20 Standard spezifiziert Frequenzen unterhalb von 3,5 GHz. Es soll eine Datenübertragungsrate von 1 Mbit/s erreicht werden und das von Fahrzeugen aus, die mit einer Geschwindigkeit von bis zu 250 km/h fahren

2.2.1.2 Sicherheit von WLANs

Der erste Standard zur Sicherung von WLANs war Wired Equivalent Privacy (WEP). WEP basiert auf einem vorher vereinbarten Geheimnis zwischen Access Point (AP) und dem Endgerät, welches mit dem bzw. über den AP kommunizieren will. WEP ist als unsicher bekannt. Es gibt diverse frei im Internet verfügbare Programme, mit denen sich eine WEP Verschlüsselung brechen lässt. Zwei bekannte Beispiele hierfür sind „Airsnort“, welches unter http://sourceforge.net/project/showfiles.php?group_id=33358 frei erhältlich ist oder „Aircrack“, welches man sich unter http://www.linuxsoft.cz/en/sw_detail.php?id_item=5417 herunterladen kann. Die Programme schneiden bei Abhören des Datenverkehrs hinreichend viele schwache Initialisierungsvektoren (IVs) mit, um auf den WEP Schlüssel zu schließen. Je nach Schlüssellänge werden ca. 100.000 bis 1.000.000 der 24 Bit langen IVs benötigt. Nachdem sich WEP als unsicher herausgestellt hatte und die Verabschiedung des Standards 802.11i auf sich warten ließ, wurde mit WiFi Protected Access (WPA) ein Teil des IEEE 802.11i Standards vorweg genommen, um eine sicherer Lösung zu haben.

WPA bringt zusätzlichen Schutz durch dynamische Schlüssel, die auf dem Temporal Key Integrity Protocol (TKIP) [Wi-Fi] basieren, und bietet zur Authentifizierung von Nutzern neben Pre-Shared Keys (PSKs) das Extensible Authentication Protocol (EAP) nach Standard IEEE 802.1x an. WPA verwendet ebenfalls eine RC4-Stromchiffre, welche auch schon für WEP genutzt wurde. Im Gegensatz zu WEP nutzt WPA darüber hinaus jedoch weiter Mechanismen, wie eine „Per-Packet-Key-Mixing“-Funktion, einen „Re-Keying“-Mechanismus, und einen Message Integrity Check (MIC) namens Michael.

Die Authentifizierung über EAP wird meist in großen Wireless-LAN-Installationen genutzt, da hierfür eine Authentifizierungsinstanz in Form eines Servers (z. B. ein RADIUS Server) benötigt wird. Im Heimbereich und in kleineren Firmen werden meist

PSKs genutzt. Der PSK muss allen Teilnehmern des WLAN bekannt sein, da mit seiner Hilfe der Sitzungsschlüssel generiert wird.

Das oben erwähnte „Aircrack“ bietet auch die Möglichkeit WPA mit einem Wörterbuchangriff zu attackieren. Es liest den beim Verbindungsaufbau stattfindenden Vier-Wege-Handshake einer WPA-Verbindung mit und versucht diesen anschließend mittels Brute-Force zu entschlüsseln.

WPA2 ist der Nachfolger von WPA. Es ist der momentan aktuelle Sicherheitsstandard für Funknetze nach IEEE 802.11a, b und g. WPA2 implementiert die grundlegenden Funktionen von IEEE 802.11i. Im Unterschied zu WPA kann bei WPA2 der Advanced Encryption Standard (AES) anstatt RC4 zur Verschlüsselung eingesetzt werden. Zusätzlich zu TKIP kommt bei WPA2 noch das „Counter Mode with Cipher Block Chaining Message Authentication Code Protocol“ (CCMP) zum Einsatz. CCMP basiert auf dem Advanced Encryption Standard (AES) und erlaubt sowohl Verschlüsselung als auch Integritätsprüfung der Daten. Es soll TKIP langfristig ersetzen.

Die Sicherheit von WLAN wird nach [BSI06] wie in Tabelle 3 folgt zusammengefasst. Dabei werden die dem „state of the art“ entsprechenden Mechanismen berücksichtigt.

Tabelle 3 Bewertung der Elemente von IEEE802.11i bzw. WPA und WPA2[BSI06]

Funktion	**Verfahren**	**Bewertung**	**Kommentar**
Authentisierung	implizite Authenti-sierung durch Pre-Shared Key	0	Diese Bewertung gilt, sofern der Schlüssel zufällig gewählt ist bzw. aus einem Passwort hoher Komplexität mit einer Länge von mindestens 20 Zeichen erzeugt wird
	IEEE 802.1X	++	Schlüsselmanagement und diverse Authentisie-rungmethoden werden unterstützt. Die verwendete Authentisierungmethode muss dem zu erreichenden Sicherheitsniveau angemessen gewählt sein. Nur für diesen Fall gilt die angege-bene Bewertung.
Verschlüsselung (WPA)	TKIP	+	TKIP basiert auf WEP. Es erfolgt für jedes Paket eine kryptographische Erzeugung eines Schlüssels. Da TKIP in Software abläuft, kommt es zu Leistungseinbußen.
Integritäts-prüfung (WPA)	Michael	0	DoS-Angriff ist möglich. Die Länge des MIC beträgt 64 Bit.
Verschlüsselung (WPA2)	CCMP	++	CCMP verwendet AES. AES erfordert entspre-chende Hardware. Die verwendete Schlüssel-länge beträgt 128 Bit. Nach dem Stand der Technik ist CCMP als sicheres Verfahren einzustufen.
Integritäts-prüfung (WPA2)	CBC-MAC	++	Bestandteil von CCMP. Die Länge des MIC beträgt 64 Bit.
Legende: "++" = sehr gut, "+" = gut, "0" = akzeptabel, "-" = mangelhaft, "--" = ungenügend			

Der Autor dieser Arbeit schließt sich dem BSI in der Bewertung an.

2.2.2 WiMAX

Parallel zu IEEE 802.11 für WLAN entwickelt das IEEE den IEEE 802.16 Standard für „Broadband Wireless Access“ bzw. ein drahtloses MAN. Beim European Telecommunication Standards Institute (ETSI) ist der entsprechende Standard unter dem Namen „Worldwide Interoperability for Microwave Access“ kurz WiMAX oder auch Hiper-MAN bekannt.
WiMAX ist eine Funktechnologie, welche Wireless-LAN ähnelt. Die erste Standardisierung im Dezember des Jahres 2001 legt die Funkschnittstelle für die Richtfunk-Frequenzen zwischen 10 MHz und 66 MHz fest. Diese Technik sollte ursprünglich eine Reichweite von 50 Kilometer bei 70 MBit/s Übertragungsrate erreichen. Dies gilt allerdings nur im Idealfall, d.h. bei direkter Sichtverbindung und stationären Empfängern. Die Tatsache, dass in diesem Frequenzbereich für die Übertragung aufgrund der Ausbreitungseigenschaften der elektromagnetischen Wellen eine freie Sichtverbindung mit Außenantennen notwendig ist, was für mobile Anwendungen unzweckmäßig ist, hat dazu geführt, dass danach auch der darunter liegende Frequenzbereich zwischen 2 GHz und 11 GHz ohne Sichtverbindung zwischen Sender und Empfänger hinzugenommen und Anfang 2003 in dem IEEE Standard 802.16a festgehalten wird. Damit sind auch Indoor-Anwendungen, wie etwa Einsteckkarten für Notebooks möglich geworden und das Thema wird damit auch wirtschaftlich interessant. Im Jahr 2004 werden dann beide Standards überarbeitet und als 802.16d-2004 bzw. "Wimax fixed" zusammengefasst.
In der Untergruppe 802.16e wurde an einer Erweiterung gearbeitet, die für mobile Anwendungen bei niedrigen Geschwindigkeiten in lizenzfreien und lizenzpflichtigen Frequenzspektrum unterhalb von 6 GHz arbeitet. Dieser Standard wird "Wimax mobile" genannt. Unabhängig vom IEEE hat Südkorea mit einem Projekt namens „Wireless Broadband“, kurz WiBro eigene Schritte unternommen. Ab 2004 arbeitet Südkorea dann an der IEEE-Standardisierung mit. Deshalb fließt WiBro erst unter 802.16e in den Standard ein.
Für den Standard "WiMAX fixed" sind für den Frequenzbereich bis 11 GHz auf der physikalischen Schicht drei Übertragungsarten definiert:

- Einträger-Modulation
- Orthogonal Frequency Division Multiplexing mit 256 Trägern kurz 256-OFDM[10]

[10] Zum besseren Verständnis des oberen Abschnitts sei hier der Begriff OFDM erläutert:

OFDM steht für Orthogonal Frequency Division Multiplex. OFDM ist ein Vielfachträgerverfahren. Es verteilt den Datenstrom auf viele schmale Träger innerhalb der Kanalbandbreite. Kommt es innerhalb des Frequenzspektrums zu Störungen, dann ist nicht der gesamte Datenstrom betroffen, sondern nur ein einzelner Träger. Im Vergleich zu einem Einträgerverfahren müssen im Störungsfall weniger Daten wiederholt übertragen werden. OFDM zeichnet sich deshalb als äußerst robust aus. Mit Equalizern und Korrekturfunktionen lässt sich bei OFDM eine festgelegte Übertragungsrate sicherstellen. Die Umwandlung des Signals von Frequenz- in den Zeitbereich bzw. umgekehrt mittels Fast-Fourier-Transformation kann man leicht in Hardware als fest programmierte Funktion implementieren. Orthogonal Frequency Division Multiplex Access kurz OFDMA ist eine Variante von OFDM. Diese Variante ermöglicht es den Sub-Trägern unterschiedliche Nutzer zuzuweisen. Dadurch können mehrere unterschiedliche Betreiber eine Basisstation teilen.

- Orthogonal Frequency Division Multiplexing mit 2048 Trägern kurz 2048-OFDM

Das WiMAX-Forum favorisiert 256-OFDM. Bei WiMAX arbeitet 256-OFDM mit 256 Trägern und Kanalbandbreiten von 1,25 bis 20 MHz. In diesem 20 MHz breiten Kanal mit einer Modulationseffizienz von 5 Bit/s pro Hertz überträgt das System 100 MBit/s. Während „WiMAX fixed“ für Funkverbindungen deren Stationen sich nicht bewegen, feste Verbindungen also, entworfen wird, ist „WiMAX mobile“ für tragbare Geräte, wie z.B. Notebooks verwendet werden. Unsicher ist, ob später "WiMAX mobile" mit "WiMAX fixed" kompatibel ist. WiMAX deutet auf Kompatibilität hin, die in Wirklichkeit nicht vorhanden ist. Im Prinzip handelt es sich um zwei unterschiedliche Versionen. Ursprünglich sollte "WiMAX mobile" auch 256-OFDM als Übertragungsart nutzen. Durch die Bemühungen von Südkorea ist noch die Übertragungsart Scalable OFDMA kurz SOFDMA mit 2048 Trägern pro Kanal hinzugekommen. SOFDMA unterstützt mobile Anwendungen durch Sub-Chanelization bzw. Untergruppenbildung besser. In einer Kombination aus TDMA und OFDMA erlaubt SOFDMA den Zugriff auf die Untergruppen der Frequenzträger.
Aufgrund der vielen verschiedenen Arten von Zugriffsverfahren auf die Funkschnittstelle ist die Kompatibilität innerhalb des Standards gefährdet. Die Gefahr für WiMAX ist die Möglichkeit von unterschiedlicher Hardware, die untereinander nicht kompatibel ist. Hardware mit einem „Kombo“-Chipsatz, der mehrere Standards beherrscht ist zwar eine Möglichkeit aber unverhältnismäßig teuer.

Da im Rahmen der in Kapitel 9 beschriebenen Implementierung ein WiMAX Board von Fujitsu zu Verfügung steht, wird im Folgenden auf die Sicherheit von WiMAX, auf dem zu Verfügung gestellten Board nach [Nück06] eingegangen[11]:

Wie viele andere Standards enthält die WiMAX Spezifikation eine Teilschicht für die Sicherheit, welche die erforderliche Infrastruktur und die Protokolle für die Authentifizierung und Verschlüsselung festlegt. Dabei sind zwei Punkte zu betrachten:

1. Die Handhabung der Sicherheit für Anwendungen eines Endnutzers.
2. Die Sicherheitsmaßnahmen zwischen einer Teilnehmerstation (SS) und einer Basisstation (BS), um verschlüsselt über das drahtlose Breitbandnetz kommunizieren zu können.

Während der erste Punkt durch die Installation eines geeigneten Softwarepaketes wie z. B. Pretty Good Privacy (PGP) [PGP] erfüllt werden kann, muss der zweite Aspekt vom Standard definiert werden. Man kann sagen, der Unterschied liegt darin, dass zwei Nutzer Systemsicherheitsprotokolle unabhängig vom spezifischen Datentransportnetz nutzen können. Im zweiten Fall ist es z. B. einer BS möglich, die Identität eines SS zu verifizieren, indem sie die Sicherheitsprotokolle nutzt, wie diese im Standard definiert sind. Die im WiMAX Standard definierten Protokolle bieten Sicherheit für Nutzer im Netz.

[11] Um Fehler durch eine Übersetzung zu vermeiden, wird der englische Originalwortlaut übernommen.

Der Schutz der Privatsphäre basierte auf zwei Aspekten. Der erste gibt an, welche kryptographischen Protokolle benutzt werden und wie sie auf die Nutzdaten angewandt werden. Der zweite Aspekt betrifft die Handhabung und Generierung von Schlüsseln. Der WiMAX Standard liefert folgendes Protokoll:

Ein SS ist mit einem digitalen Zertifikat entsprechend dem X.509 Standard ausgerüstet. Ein solches Zertifikat enthält außer zusätzlichen Informationen einen SS spezifischen privaten Schlüssel, der z.B. vom Hersteller des SS generiert ist. Dieses Zertifikat wird zur BS gesandt, und der öffentliche Schlüssel wird benutzt, einen gemeinsamen Schlüssel zu erzeugen. Der SS kann nun den von der BS erzeugten Schlüssel durch Nutzen seines privaten Schlüssels dechiffrieren Als Folge haben BS und SS einen Schlüssel verfügbar, der in einem symmetrischen Chiffrierschema benutzt werden kann. Die auf RSA basierenden asymmetrischen kryptographischen Protokolle nutzen Schlüsselgrößen zwischen 1024 und 2048 bits.

Drei symmetrische kryptographische Funktionen werden im Standard benutzt:

1. Der Data Encryption Standard (DES)
2. Der Triple-DES (3DES)
3. Der Advanced Encryption Standard (AES)

Um jedes der definierten Protokolle unterstützen zu können, ist WiMAX SoC MB87M3400 mit zwei dedizierten Hardewareblocks für DES und AES ausgestattet wie in Abbildung 7 dargestellt.

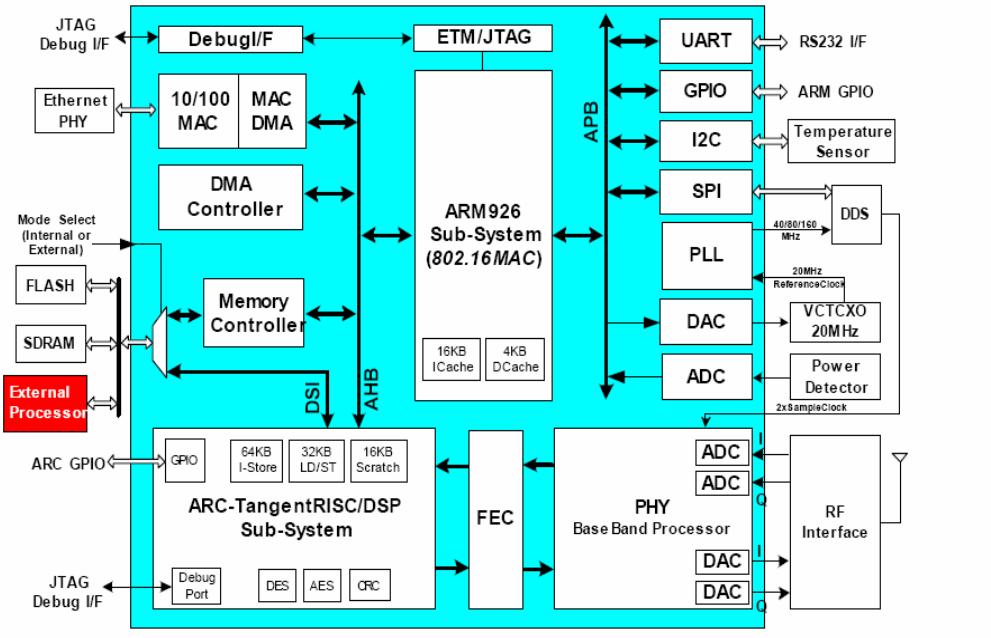


Abbildung 7: The MB87M3400 WiMax SoC architecture [Nück06]

Für Anwendungen wie Echtzeitvideokonferenzen mit vertrauenswürdigen Partnern wird eine passende Kombination von Software, Hardware und eingesetzten Protokollen

benötigt. Der 802.11-16-2004 WiMax Standard bietet einen DES/AES Hardwareblock an, um die Sicherheitsanforderungen effizient und kostengünstig zu ermöglichen. Zusätzlich liefert der benutzte ARM 926 Prozessor genügend Leistung, die Schlüsselaustauschalgorithmen auszuführen. Alternativ kann ein optionaler externer Prozessor, mit z. B. einem 600 MHz Power PC, mehr Rechenleistung liefern. Diese kommerziell verfügbare Konfiguration wird für BS Anwendungen typischerweise benutzt, wenn eine große Zahl Nutzer bedient werden müssen. Weiter unterstützt das externe Speichersystem eine Flashspeichernutzung für "stand alone booting" und für die Speicherung der Konfiguration und Schlüsselinformation

2.2.2.1 Anwendung

WiMAX hat als Grundidee den Ersatz von breitbandigen Kabelnetzen, wie etwa DSL, durch eine Funkübertragung zu ersetzen. Daher wird es auch als W-DSL bzw. Wireless-DSL bezeichnet. Im Gegensatz zu WLAN ermöglicht WiMAX einen erheblich größeren Durchmesser des Versorgungsbereichs einer Basisstation. Mehrere Kilometer Reichweite lassen die letzte Meile zwischen Netzbetreiber und Kunden schrumpfen. Vor allem in Gegenden, wo DSL oder Kabel keinen Internet-Zugang bieten können, ist WiMAX eine Alternative. Mobilfunkbetreiber können ihre Mobilfunk-Basisstationen über Funk mit einer dem Festnetz vergleichbaren Zuverlässigkeit anbinden. So lassen sich teure Kabelverbindungen sparen. Typische Anwendungen sind auch drahtlose Standleitungen zur Standortvernetzung von Firmen, bei denen WLAN von der Reichweite nicht ausreicht und der klassische Richtfunk zu teuer ist.
Der Einsatz von WiMAX-Netzen scheint hauptsächlich in solchen Ländern interessant, deren großflächige Städte wegen einer geringen Dichte in der Telefonversorgung noch nicht so engmaschig verkabelt sind wie in Europa, also Länder wie China, Russland, Indien oder die Türkei. In hoch industrialisierten Ländern wird diese Technik auch noch Schwierigkeiten zu überwinden haben, die passenden Funklizenzen zu bekommen und Standorte für die Antennenstationen, die eine recht hohe Sendeleistung benötigen, zu finden.
Somit gibt es nicht "das" WiMAX-Produkt, sondern eine ganze Palette unterschiedlicher Lösungen, eine kurz gefasste Beschreibung erschwert und die Gefahr von Missverständnissen erhöht:
Die folgende Abbildung 8 gibt einen Überblick über die Anwendungsbereiche von WiMAX.

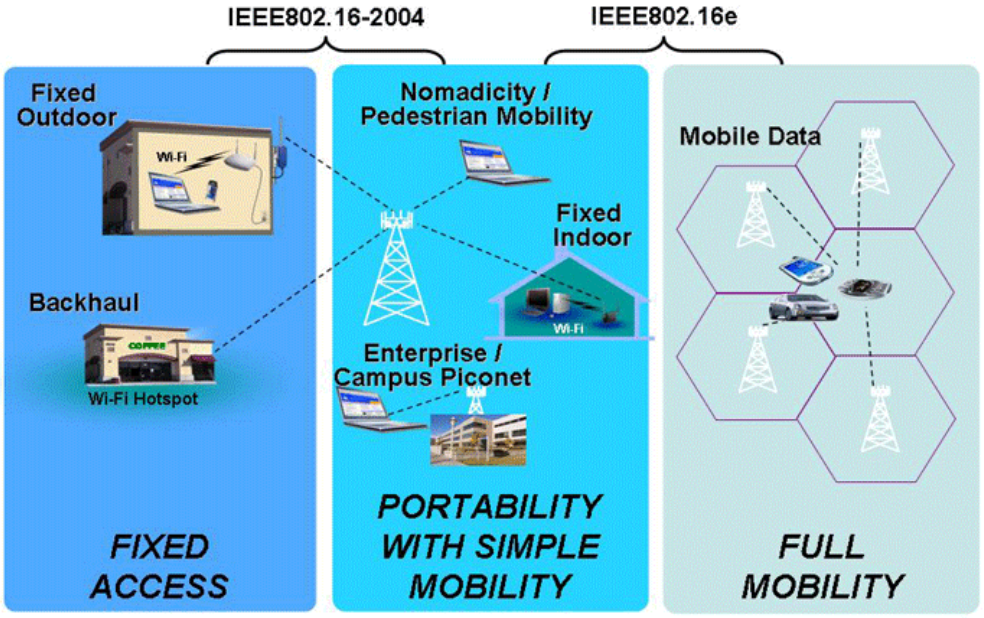


Abbildung 8: Überblick über WiMAX [Quelle: Intel]

Was die Sicherheit angeht ist zum gegenwärtigen Zeitpunkt davon auszugehen, dass WiMAX dieselben Mechanismen zu Verfügung haben wird, wie WLAN.

2.2.3 PHS

Das „Personal Handyphone System“ (PHS) ist ein 1989 in Japan entwickelter Mobilfunkstandard [Walke01a]. Ein zu DECT vergleichbarer Standard existierte zu dieser Zeit in Japan noch nicht. Es sollte vor allem ein im Vergleich zu den konventionellen zellularen Mobilfunksystemen kostengünstiger neuer Standard entwickelt werden.

PHS wurde sowohl für den privaten als auch den öffentlichen Bereich konzipiert. Eine Personal Station kurz PS unterstützt als PHS-Endgerät daher auch zwei entsprechende Betriebsmodi. Es kann je nach Verfügbarkeit zwischen den beiden Modi entweder per Hand oder automatisch hin- und hergeschaltet werden. Dieser Vorgang stellt aber kein Handover während eines Gespräches dar. Optional gibt es auch einen Dualmode bei dem ein PS sowohl einen Ruf von einem öffentlichen als auch privaten Netz empfangen kann.

Wie auch DECT verwendet PHS ein hybrides TDMA/FDMA-Schema mit TDD im Frequenzbereich von 1893,5 bis 1919,6 MHz. Es gibt inzwischen auch ein „Advanced PHS“, welches eine Übertragungsrate von 1 MBit/s zulässt und zu den 3G Netzen gezählt wird.

Die PS kennt verschiedene Authentifizierungsmechanismen [PHSa]. Beim “Roaming“ wird die PS immer vom besuchten Netz, unter Verwendung von Informationen, welche von ihrem Heimnetz zum besuchten Netz kopiert werden, authentifiziert. Das PHS kennt auch “call termination authentication”. Hierbei soll sichergestellt werden, dass die Verbindung zur richtigen Person getrennt wird [PHSb]. Handover in PHS wird in [PHS97, PHS_HO99] genau beschrieben. Neben den Authentifikationsmöglichkeiten bietet PHS analog zu dem im nächsten Abschnitt beschriebenen „Digital European Cordless Telecommunications“ kurz DECT auch die Möglichkeit der Verschlüsselung.

2.2.4 DECT

Das Europäischen Standardisierungsinstitut für Telekommunikation kurz ETSI hat 1992 den „Digital European Cordless Telecommunications“ kurz DECT –Standard [DECT] ETS 300 175 festgelegt [Walke01b, Schiller03]. Dieser Standard ist im Haus- und Firmenbereich stark verbreitet. Mittlerweile sind etwa 300 Millionen DECT-basierte Systeme weltweit installiert.
Die Regulierungsbehörden von Australien, Hongkong, China und den USA standardisierten DECT für ihre Gebiete, teilweise in anderen Frequenzbereichen. Inzwischen sind DECT-Varianten in über 100 Ländern im Einsatz.[12] Von den Industrieländern stehen nur Japan und Südkorea abseits.
Frequenzbereich: DECT wird in den meisten Ländern in einem speziell freigegebenen Frequenzbereich betrieben. Dieser liegt in Europa zwischen 1880 und 1900 MHz. Auf anderen Kontinenten werden teilweise auch andere Frequenzbereiche von 1,5 GHz bis zu 3,6 GHz verwendet.

Die Verteilung der Frequenzen für die verschiedenen Kanäle innerhalb dieses Frequenzbandes folgt einem MC/TDMA/TDD Algorithmus. Die Reichweite der DECT-Systeme ist in Gebäuden auf etwa 50 m beschränkt. Im Freien können bis zu 300 m erreicht werden. DECT unterstützt in seiner grundlegenden Spezifikation die synchrone und symmetrische Übertragung von Sprache. Eine Ergänzung liefert wichtige Dienste für die paketorientierte Datenübertragung. Unter Ausnutzung aller Kanäle stehen maximal 20 Mbps Datentransferrate zur Verfügung.

Unbefugte Benutzung und unbefugtes Mithören wird bei DECT wie bei anderen Mobilfunksystemen auch durch drei Methoden verhindert:

- Anmelden: Der mobile Teilnehmer meldet der Basisstation seine Empfangsbereitschaft.
- Ausweisen: Bei jedem Rufaufbau muss sich das Mobilgerät bei der Basisstation durch Verwendung eines geheimen Schlüssels ausweisen. Verschlüsseln: Die Nutzdaten (Sprache oder Daten) werden während der Funkverbindung kodiert und auf der Gegenseite dekodiert, wobei ein Schlüssel verwendet wird, der beiden Gegenstellen bekannt ist, aber selbst nicht über Funk übertragen wird. Der verwendete Verschlüsselungsstandard nennt sich DECT Standard Cipher.

[12] Da es sich bei DECT nun nicht mehr um einen rein europäischen Standard handelt, wurde das Wort „European“ in der Definition von DECT durch das Wort „Enhanced“ ersetzt.

- Die Verschlüsselung ist ein optionaler Teil der DECT-Spezifikation und wird nicht von allen Geräten unterstützt.

Die französische "Direction Centrale de la Sécurité des Systemès d'Information" (DCSSI) bietet schon 2004 einen Kurs „cryptanalyse opérationnelle du DECT“ an. Dies lässt vermuten, dass die DCSSI die Echtzeitentschlüsselung von DECT beherrscht, und somit in der Lage ist, verschlüsselte Telefonate mit DECT-Telefonen abzuhören. Es verwendet den 'DECT Standard Cipher', einen Stromverschlüsselungsalgorithmus zur Verschlüsselung der Nutzdaten. DSC wurde von der ETSI standardisiert. die Spezifikation wird jedoch nur an handverlesene Firmen abgegeben. Möglich ist auch, dass in Frankreich eine geschwächte Variante von DECT zum Einsatz kommt. Aufgrund dieser Unklarheiten wird DECT hier als prinzipiell unsicher angesehen und nicht tiefergehend behandelt.

2.2.5 IS-95

Der amerikanische IS95 Standard ist das Analogon zu GSM.

Ein so genannter A-Schlüssel, das ist ein 64 Bit Schlüssel, wird im Endgerät und in dem zugehörigen Authentication Center (AC) gespeichert. Bei IS-95 findet der Cellular Authentication and Voice Encryption (CAVE) Algorithmus sowohl zur Verschlüsselung der Übertragung als auch zur Authentifikation Verwendung. Zur Authentifikation wird hier wie üblich ein Challenge Response Verfahren verwendet.

Das AC generiert zunächst eine 56-bit Zufallszahl RANDSSD. Unter Benutzung des CAVE Algorithmus werden dann SSD-A and SSD-B berechnet. Der Zufallswert RANDSSD und die berechneten SSDs werden vom AC zur Basisstation (BS) geschickt. Diese sendet eine UPDATE_SSD_ORDER Nachricht, welche den RANDSSD beinhaltet zum Endgerät. Das Endgerät berechnet mit dem CAVE Algorithmus die SSD. Dann generiert es einen Zufallswert (RANDBS) um AUTHBS zu berechnen. Das Endgerät schickt RANDBS in der BASE_STATION_CHALLENGE_ORDER Nachricht zur BS. Die BS berechnet dann AUTHBS und Schickt diesen Wert in der BASE_STATION_CHALLENGE_CONFIRMATION Nachricht zurück zum Endgerät. Wenn der vom Endgerät berechnete AUTHBS-Wert mit dem von der BS gesendeten Wert übereinstimmt, antwortet das Endgerät mit einer UPDATE_SSD_CONFIRMATION Nachricht. Das stellt sicher, dass das Endgerät und die BS über dieselbe SSD verfügen. SSD-A wird dann für die Authentifikation verwendet und SSD-B für die Verschlüsselung der Übertragung.

2.2.6 Bluetooth

Die Bluetooth [Bluetooth, Blue01, Schiller03] Überträger arbeiten im ISM (Industrial, Science, Medical) Frequenzband. In den meisten Ländern liegt dieses Band im Frequenzbereich von 2400 MHz bis 2483,5 MHz. Die Übertragung erfolgt im Fast Frequency Hopping Verfahren. Dabei wird das zur Verfügung stehende Frequenzband in 79 Kanäle geteilt, zwischen denen 1600 Mal in der Sekunde gewechselt wird.

Gegenüber anderen Verfahren, wie z.B. dem DECT Standard, bei denen nur etwa ein Frequenzwechsel pro Sekunde erfolgt, hat das Fast Hopping drei entscheidende Vorteile:

- Größere Unempfindlichkeit gegenüber Störstrahlung: Ist ein Frequenzband durch eine Störstrahlung wie z.B. von einem Mikrowellenherd verursacht blockiert, ist nur für eine 1/1600 Sekunde die Übertragung unterbrochen, die Performance der Funkverbindung beträgt dann immer noch 1599/1600 bzw. 99,93%, wenn tatsächlich nur ein Kanal betroffen ist.
- Höhere Sicherheit: Die schnellen Frequenz- Wechsel und der dahinter liegende Algorithmus sind nur mit sehr aufwendiger Technik zu erfassen und kaum zu decodieren.
- Dominant gegenüber anderen Funkverbindungen: Bluetooth kann sich gegenüber anderen Funkverbindungen durchsetzen: Trifft eine DECT Modul auf ein blockiertes Frequenzband, wartet es eine Sekunde, bis der Sprung auf das nächste Band erfolgt. Da Bluetooth 1600 Mal pro Sekunde springt, ist auch die Wahrscheinlichkeit, dass Bluetooth eine DECT-Übertragung blockiert 1600 Mal höher, als dass DECT Bluetooth stört.

Einen Überblick über die Sicherheit von Bluetooth gibt [Anand01, Carter00]. Bluetooth kennt drei Sicherheitsmodi.

- In Modus 1 werden die Sicherheitsfunktionen der Sicherungsschicht komplett ignoriert. Dieser Modus ist für unkritische Anwendungen, wie z.B. den Austausch von Visitenkarten gedacht.
- Modus 2 bietet Sicherheit auf der Dienstebene und ermöglicht speziell bei parallel ablaufenden Anwendungen mit unterschiedlichen Sicherheitsanforderungen vielseitigere Zugriffsverfahren.
- Modus 3 bietet Sicherheit auf der Sicherungsebene durch die der Verbindungsmanager für alle Anwendungen bei Verbindungseinrichtung für Sicherheitsvorkehrungen auf einem gemeinsamen Niveau sorgt. Modus 3 sorgt für ein gemeinsames Sicherheitsniveau ist aber weniger flexibel als Modus 2.

Der Bluetooth-Standard ermöglicht es zwei Geräten prinzipiell miteinander sicher zu kommunizieren. Hierfür ist allerdings eine Initialisierungsprozedur - Pairing oder Kopplung genannt - notwendig. Bei der Kopplung wird bei beiden Geräten ein maximal 16 Bytes langer Code eingegeben. Es wird bei dieser Prozedur davon ausgegangen, dass dies in einer sicheren Umgebung auf sichere Art und Weise erfolgt, was bedeutet, dass keine unbefugten Dritten in der Lage sind, an den Code zu gelangen, wie es z. B. der Fall wäre, wenn Sie dem Nutzer bei der Eingabe des Codes über die Schulter schauten. Wenn dies der Fall ist, dann können die beiden gekoppelten Geräte sich sicher authentifizieren und verschlüsselt miteinender kommunizieren. Allerdings wird der Code bei den meisten Implementierungen auf eine vierstellige PIN reduziert. Des Weiteren ist anzumerken, dass gerade Bluetooth ein gutes Beispiel dafür ist, dass eine prinzipiell sichere Technologie auf Grund von Implementierungsfehlern ein Gerät unsicher machen kann. BlueSnarf ist die bekannteste Attacke, die das ausnutzt. Über das OBEX Push Protokoll können hier mittels eines „OBEX-Get“ Zugriff auf Dateien auf dem Endgerät erlangt werden, wie z.B. das Adressbuch und den Kalender. Betroffen sind davon weitverbreitete Geräte, wie das

T68i, das T610 und das T630 als Handys von Sony-Ericson und die Nokia-Modelle 6310 und 6310i bei entsprechenden Firmware Versionen. Unter http://linux.softpedia.com/get/Security/Bluediving-8443.shtml kann man sich das Tool Bluediving herunterladen welches Bluesnarf und andere Angriffe implementiert.[13] Auch wenn kein Implementierungsfehler vorliegt, muss man feststellen, dass die sicheren Modi von Bluetooth sich für Roaming nicht eignen, da jedes Mal eine Nutzerinteraktion beim Koppeln erforderlich ist, was im Widerspruch zu der Möglichkeit eines übergangslosen Netzwechsels steht.

2.2.7 GSM

Die „Groupe Speciale Mobile“ wurde von der „Conférence Européenne des Administrations des Postes et des Télécommunications“ kurz CEPT - der Europäischen Konferenz der Verwaltung für Post und Fernmeldewesen also - ins Leben gerufen. Sie entwickelte das nach ihr benannte Mobilfunksystem GSM [Walke01a, Schiller03]. Die Standardisierung von GSM wurde bei CEPT begonnen, vom ETSI dem Europäischen Institut für Telekommunikationsnormen weitergeführt und später an das „3rd Generation Partnership Project“ kurz 3GPP übergeben. Dort wird GSM unter dem Begriff GERAN, der von „GSM EDGE Radio Access Network“ herrührt weiter standardisiert. Die Sicherheitseigenschaften von GSM sind in [GSM01a, GSM01b] aufgeführt und im Detail z. B. auch in [Van96] beschrieben. [33.102, 33.103, 33.105] beschreiben die Sicherheitsarchitektur, Integrationsrichtlinien und die Anforderungen an die Algorithmen bei GSM.

Die Authentifikation erfolgt normalerweise beim Einbuchen und beim Handover. Zunächst authentifiziert der Benutzer sich gegenüber der SIM-Karte in seinem Endgerät mittels Eingabe der PIN. Anschließend authentifiziert sich die SIM-Karte im Endgerät gegenüber der Basisstation in einem Challenge-Response-Protokoll.

Hierbei sendet zunächst das Mobilfunksystem eine vom Authentifikationszentrum (AC) erzeugte 128 Bit lange Zufallszahl RAND an die SIM-Karte.
Unter Nutzung des geheimen Algorithmus A3 [Walke01a] und des geheimen Schlüssels K_i generiert die SIM-Karte den 32 Bit langen Wert SRES = A3(RAND, K_i) und sendet diesen zurück. Der Netzbetreiber berechnet mittels RAND und K_i aus seiner AC den Wert SRES’=A3(RAND, K_i). Wenn SRES=SRES’ gilt, dann ist der Teilnehmer authentifiziert.
Wichtig ist hierbei, dass der geheime Teilnehmerschlüssel K_i niemals über die Luft-Schnittstelle übertragen wird.

Der Nutzer wird authentifiziert. Er kann jedoch selbst das Netz nicht authentifizieren. Dies hat zur Folge, dass sich ein mobiles Endgerät gegenüber jedem Anfrager authentifiziert – egal ob autorisierte Basisstation oder Angreifer. Der Nutzer kann sich nicht sicher sein, mit welchem Netz er kommuniziert. Dies ist ein Sicherheitsproblem. So

[13] Der Autor dieser Arbeit hatte im Rahmen seiner Tätigkeit beim Fraunhofer Institut SIT Gelegenheit bei der Betreuung des SIT-Standes auf der ETRICS2006 Konferenz in Freiburg dies Konferenzteilnehmern vorzuführen bzw. es mit verschiedenen Endgeräten verschiedener Teilnehmer auszuprobieren.

genannte IMSI[14]-Catcher nutzen dies aus und zwingen das mobile Endgerät, seine IMSI an sie zu übertragen. Dies ermöglicht u. a. die Erstellung von Bewegungsprofilen des Nutzers des Endgerätes, was einen Eingriff in die Privatsphäre darstellt und Überwachungsmöglichkeiten eröffnet.

Im GSM-Standard ver- bzw. entschlüsselt der Algorithmus A5 die über die Luftschnittstelle übertragenen digitalisierte Sprachdaten zwischen einem GSM-Endgerät und einer Basisstation. Er wurde bereits 1987 vorgestellt. Zwei Versionen des A5 Algorithmus sind bekannt. Die kryptographisch stärkere Version wird mit A5/1 bezeichnet. Der Algorithmus A5/2 gilt als schwaches Verschlüsselungsverfahren. A5/2 ist für den Export in Länder konzipiert, in denen der Einsatz von Kryptographie von Staatswegen nur unter Auflagen zulässig ist. Die Schlüssellänge wurde von 64 Bit bei A5/1 auf 16 Bit bei A5/2 reduziert. A5/2 wurde aufgrund seiner Schwäche von der 3GPP ab Release 6 aus dem Standard herausgenommen. A5 ist ein schneller Stromchiffrieralgorithmus. Er besteht aus drei Schieberegistern mit linearer Rückkopplung (LFSR), die. eine Länge von 19 Bits, 22 Bits und 23-Bits haben. Addiert man die Bitanzahl der drei Register erhält man die Schlüssellänge von 64 Bit.

Bei Untersuchungen von Handys verschiedener Hersteller wurde festgestellt, dass bei allen untersuchten Implementierungen zehn dieser 64 Bits auf "0" gesetzt sind. Die Schlüssellänge wird somit auf effektiv 54 Bit reduziert [BGW99].

Für die Vertraulichkeit der Gespräche ist der A5-Algorithmus entscheidend. Da der A5 eines der meistgenutzten Verschlüsselungsverfahren der Welt ist, darf man davon ausgehen, dass er von den Geheimdiensten und anderen Organisationen ausführlich analysiert wurde. Bei der Internetveröffentlichung schlug der englische Kryptograph Anderson einen Angriff der Komplexität 2^{40} vor [Ande94b]. Eine noch gefährlichere Attacke stellte Golic vor. Durch einen "time-memory-trade-off"-Angriff kann A5 leicht gebrochen werden [Goli97].

Die ETSI hat als GSM-Standardisierungsgremium einen neuen Algorithmus A5/3 veröffentlicht. Hierbei handelt es sich um den auch bei UMTS verwendeten Blockchiffrieralgorithmus KASUMI [35.202]. Die Spezifikation von KASUMI wurde aus dem japanischen Misty-Algorithmus entwickelt.
Misty ist ein Blockalgorithmus mit einer Schlüssellänge von 128 Bit, einer Blockgröße von 64 Bit und 8 Runden (Misty1) bzw. 12 Runden (Misty2), der von H. Ohta und M. Matsui 1997 vorgestellt wurde. Dieser Algorithmus wurde entwickelt, um der differentiellen sowie der linearen Kryptoanalyse zu widerstehen und kann sowohl in Hard- als auch in Software effizient eingesetzt werden. Es sind bis jetzt keine erfolgreichen kryptoanalytischen Angriffe bekannt geworden. Die Sicherheit von Misty ist in [Nessie] ausführlich betrachtet: Misty bietet danach „provable security against differential and linear cryptanalysis.“

[14] IMSI steht für International Mobile Subscriber Identity

A5/3 soll nicht nur die GSM-Sprachtelefonie, sondern auch die GSM-Datenprotokolle GPRS, HSCSD und EDGE absichern. Dies stellt insbesondere bei GPRS eine Verbesserung dar, da der bisher dort eingesetzte GPRS Encryption Algorithm (GEA) nie veröffentlicht wurde, und dementsprechend nicht bedenkenlos als sicher oder vertrauenswürdig eingestuft werden kann.

2.2.8 GPRS

Der General Packet Radio Service (GPRS) [Dixit02, Schiller03, Walke01a] ist ein Standard, der eine Übertragungsrate von bis zu 150 kilobits per Sekunde erlaubt. Dies bedeutet eine deutliche Verbesserung im Vergleich zu GSM mit 9.6 kilobit pro Sekunde.

GPRS ist eine Erweiterung des GSM-Mobilfunk-Standards um paketorientierte Datenübertragung. Diese wird manchmal auch in Anlehnung an 3G als 2,5G bezeichnet. Im Gegensatz zum leitungsvermittelten Datendienst High Speed Circuit Switched Data (HSCSD) ist GPRS paketorientiert, d. h., die Daten werden beim Sender in einzelne Pakete umgewandelt, als solche übertragen und beim Empfänger wieder zusammengesetzt. Die GPRS-Technik ermöglicht bei der Bündelung aller 8 GSM-Zeitschlitze eines Kanals theoretisch eine Datenrate von 171,2 kBit/s. Im praktischen Betrieb ist die Anzahl der parallel nutzbaren Zeitschlitze jedoch durch die Fähigkeit des Mobilgerätes (multislot capability) und der Netze begrenzt. Am Markt befinden sich z. Zt. Geräte mit max. vier Zeitschlitzen im Downlink und max. zwei Zeitschlitzen im Uplink Diese stehen jedoch nicht gleichzeitig zu Verfügung. Die damit erreichbare Datenrate beträgt abhängig vom verwendeten Coding Scheme und der von der Netzauslastung abhängigen Anzahl der zugeteilten Zeitschlitze bis zu 57,6 kBit/s.
Wenn GPRS aktiviert ist, besteht nur virtuell eine dauerhafte Verbindung zur Gegenstelle (sog. Always-on-Betrieb). Erst wenn wirklich Daten übertragen werden, wird der Funkraum in dem erforderlichen Zeitraum benutzt. Deshalb wird anders als bei HSCSD kein Funkkanal dauerhaft für einen Benutzer reserviert. GPRS-Abrechnungen sind deshalb hauptsächlich von den übertragenen Datenmengen abhängig, und nicht primär von der Verbindungsdauer. Der paketvermittelnde Dienst GPRS benötigt im Mobilfunknetz im Vergleich zu GSM weitere Netzelemente, den Serving GPRS Support Node (SGSN) und den Gateway GPRS Support Node (GGSN). Während der SGSN für die Mobilfunknahen Dienste wie Mobility Management und die Zugangskontrolle zuständig ist, stellt der GGSN den Übergang zum paketvermittelnden Netz, z. B. dem Internet oder einer netzinternen Diensteplattform dar.

GPRS ist aus Anwendersicht das WAN-Pendant zu WLANs; auch hier wird drahtloses Internet zur Verfügung gestellt, die zugrunde liegende Technologie ist allerdings eine andere: GPRS baut auf dem GSM-Standard auf und integriert dort die Vermittlung von IP-Paketen. Die Übertragungsraten sind mit realistischen 20–30 kBit/s allerdings verhältnismäßig niedrig, ebenso die Paketlaufzeiten von 1–2 Sekunden.

Die Sicherheit von GPRS wird in [Brookson01] ausführlich dargestellt.. Die GPRS-interne Sicherheit baut auf die Sicherheit von GSM auf [GSM03, Vin98]. Die Authentifizierung ist mit der von GSM identisch. Zu bedenken ist dabei jedoch, dass im

ungeschützten öffentlichen Internet Angriffe wie IP-Spoofing oder Connection-Highjacking möglich sind.

Der GEA Algorithmus zur Verschlüsselung der Luftschnittstelle, welcher die Vertraulichkeit gewährleisten soll, ist nicht veröffentlicht. Es dürfte sich jedoch um eine schwache Verschlüsselung mit einem etwa 40 Bit großen Schlüssel handeln, darauf deutet zumindest ein „report on specification“ [SAGE] hin.
Ein neuer, ebenfalls geheimer Algorithmus wurde nach der Lockerung der Krypto-Exportbestimmungen spezifiziert, ist aber noch nicht in Produkten umgesetzt. Selbst diese zukünftige, wohl stärkere Verschlüsselung kann jedoch durch das GSM-Netz abgeschaltet werden; der Nutzer hat also keine Kontrolle über die verwendete Sicherheit.

Schutz der übertragenen Information innerhalb des GSM-Netzes ist Sache des Netzbetreibers. Hier können also keine generellen Aussagen getroffen werden. Sobald der Datenverkehr ins öffentliche Internet wechselt, ist er so angreifbar, wie das Internet auf dieser Strecke eben ist.

2.2.9 UMTS

Die „International Telecommunication Union“ (ITU) hat unter dem Namen „International Mobile Telecommunications-2000“ (IMT-2000) in den späten 80er Jahren die Anforderungen an ein Mobilfunksystem der dritten Generation (3G-Mobilfunksystem) zusammengefasst. Das „Universal Mobile Telecommunications System“ (UMTS) erfüllt diese Anforderungen. Es gilt als Nachfolger von GSM [Roth02, Schiller03, Walke01a]. Die grundlegende UMTS-Architektur ist in [23.101] beschrieben.
Der erste wesentliche Unterschied zu GSM ist das verwendete Funkzugriffsverfahren namens Wideband- Code Division Multiple Access (WCDMA), welches auf Code Division Multiple Access (CDMA) basiert und höhere Übertragungsraten ermöglicht. Es gibt hier zwei Modi:

- Den auf Frequenzmultiplex aufbauenden „Frequency Division Duplex“ kurz FDD-Modus und
- den auf Zeitmultiplex basierenden „Time Division Duplex“ kurz TDD-Modus.

Im FDD-Modus senden das mobile Endgerät und die Basisstation in zwei verschiedenen Frequenzbereichen: Im Uplink-Kanal sendet das Mobile Endgerät, im Downlink-Kanal die Basisstation. Derzeit bauen die deutschen UMTS-Netzbetreiber ihre Netze im FDD-Modus auf, die damit erzielbare Datentransferrate liegt derzeit bei 384 kbit/s für den Downlink.
Im TDD-Modus senden das mobile Endgerät und die Basisstation im gleichen Frequenzband, jedoch zu unterschiedlichen Zeiten. Dieses Verfahren ist technisch aufwändiger. Wenn sich der Sender bewegt oder die Entfernung zur Basisstation groß ist, können Timing-Probleme auftreten. Es soll aber eine höhere Datentransferrate von bis zu 2 Mbit/s erreicht werden. Der Nutzfrequenzbereich liegt zwischen 1900 MHz und 2170 MHz

Anders als bei GSM erfolgt bei UMTS eine beidseitige Authentifikation. Ansonsten ist das eingesetzte Protokoll weitgehend kompatibel zu GSM. Es wird auch hier ein Challenge- Response-Verfahren eingesetzt. Es basiert darauf, dass ein geheimer Schlüssel K zwischen der Heimatumgebung – dem Home Environment - und dem mobilen Gerät vereinbart ist und beide die Funktionen f_1, .. , f_5, welche in [35..205, 35.206, 35.207, 35.208, 35.209] ausführlich beschrieben sind, kennen.

Das Home Environment kurz HE erzeugt für jede Verbindung eine Sequenznummer SQN und mit RAND einen 128 Bit Zufallswert. Des weiteren wird ein Authentifikationstoken unter Benutzung von K und dem operatorspezifischen 16 Bit langen Authentication Management Field kurz AMF, welches z. B: die Lebensdauer der Schlüssel festlegen kann, erzeugt. Hierzu wird zunächst ein 64 Bit MAC berechnet:

$$MAC = f_1(K, (SQN, RAND, AMF))$$

sowie der 48 Bit lange Anonymitätsschlüssel AK:

$$AK = f_5(K,RAND)$$

um dann das Authentifikationstoken AUTN wie folgt zu erhalten:

$$AUTN = SQN \text{ XOR } AK \mid AMF \mid MAC.$$

Des Weiteren werden noch die erwartete Antwort XRES und die beiden 128 Bit Schlüssel CK und IK wie folgt erzeugt:

$$XRES = f_2(K,RAND) ; CK = f_3(K,RAND) ; IK = f_4(K,RAND)$$

Mit den berechneten Daten wird schließlich der Authentifikationsvektor zusammengestellt:

$$AV = RAND, XRES, CK, IK , AUTN$$

Bei einem Verbindungsaufbau sendet das Service-Netzwerk einen Authentifikationsvektor AV an das mobile Endsystem des Benutzers. Die USIM des Benutzers muss nun auf die Challenge RAND die korrekte Response XRES berechnen. Dies geschieht wie folgt: Die USIM berechnet zunächst $AK = f_5(K, RAND)$ und extrahiert damit aus AUTN die SQN. Daraufhin wird $MAC' = f_1(K, (SQN, RAND, AMF))$ berechnet und mit dem aus AUTN erhaltenen MAC verglichen. Bei Nichtübereinstimmung sendet die USIM eine „Reject“-Nachricht an das Service-Netzwerk und die Authentifikation ist erfolglos beendet.
Bei Übereinstimmung prüft die USIM, ob SQN ein gültiger Wert ist. (Dazu verwaltet die USIM ein Feld bereits verwendeter Sequenznummern). Bei ungültiger SQN wird ein Synchronisationsfehler an das Service-Netwerk weitergemeldet.
Die USIM berechnet die Antwort $RES = f_2(K, RAND)$ und schickt diesen Wert an das Service-Netzwerk zurück. Dieses vergleicht RES mit XRES aus dem zugehörigen Authentifikationsvektor. Bei Gleichheit ist die Authentifikation gelungen.
Die USIM berechnet noch die Schlüssel CK und IK:
$CK = f_3(K,RAND) ; IK = f_4(K,RAND)$

Mittels der Daten in AUTN kann sich das HE gegenüber dem mobilen Endgerät authentifizieren, da zur Erzeugung von AUTN die Kenntnis des geheimen Schlüssels K notwendig ist. Da AUTN die zeitabhängige Sequenznummer SQN enthält würde ein wiedereinspielen erkannt werden.

Zu Verschlüsselung und zum Integritätsschutz der übertragenen Daten werden bei UMTS die Algorithmen f8 und f9 [35.201, 35.202, 35.203, 35.204] eingesetzt, welche auf dem KASUMI algorithmus im OFB und CBC Modus basieren.

2.2.10 Konklusion

In diesem Kapitel 2.2 wurde ein Überblick über die als Zugangsnetze im Rahmen des Roaming relevanten drahtlosen Netztechnologien gegeben. Dabei wurden die Sicherheitseigenschaften und die Eignung für Zugangsnetze zum Roaming analysiert.

Zusammenfassend gilt: Prinzipiell eignen sich alle auf Funk beruhenden Übertragungstechnologien für Roaming zwischen Netzen. Herauszuheben sind UMTS, WiMAX und WLAN, welche im Rahmen eines Netzwechsels eine beidseitige Authentifizierung zwischen Endgerät des Nutzers und Zugangspunkt ohne Nutzerinteraktion ermöglichen. Bluetooth ermöglicht zwar auch sichere Kommunikation zwischen zwei gegenseitig authentifizierten Geräten, allerdings ist hier eine Nutzerinteraktion erforderlich, wodurch diese Funktion beim Roaming nicht sinnvoll einsetzbar ist. Gleiches gilt auch für WUSB. Abgesehen davon sind Bluetooth wie auch DECT für den Heimbereich konzipiert, weswegen nicht zu erwarten ist, dass sie irgendwann im Rahmen von Internetzugangspunkten eine große Rolle spielen werden, vor allem wenn man bedenkt, dass WLAN Access Points auch beim Privatanwender zu Hause schon Einzug gehalten haben, da sie in Deutschland typischerweise mit einem DSL Anschluss und dem DSL Modem einhergehen. Auch geht der Trend bei Einsatz von VoIP hin zur Verwendung von WLAN und nicht von DECT [BSI06]. GSM und die übrigen Technologien sind schwächer, was die Sicherheitseigenschaften angeht, da dort keine beidseitige Authentifizierung vorgenommen wird.
WLAN bietet wie auch WiMAX sehr gute Sicherheitseigenschaften, wenn man davon ausgeht, dass alle Sicherheitsmechanismen nach IEEE 802.11i auch wirklich eingesetzt werden.
Eine SIM bzw. USIM basierten Authentifizierung ist weniger sinnvoll, da hierbei jeder Nutzer über eine entsprechende SIM-/USIM-Karte verfügen müsste, so dass auch Roaming zwischen WLANs oder WLAN und WiMAX basierten Zugangsnetzen dann für Nutzer nicht möglich wäre, die nicht Kunde eines Mobilfunkanbieters sind. Darüber hinaus wäre für die eingesetzten Endgeräte entsprechende Hardware Voraussetzung, ein weiterer Grund dafür, dass eine SIM/USIM basierte Authentifizierung nicht sinnvoll scheint, solange bei WLANs nicht standardisiert SIM oder USIM eingesetzt werden.
Da zwei der drei für Roaming wichtigsten drahtlosen Netztechnologien problemlos eine auf Zertifikaten basierende Authentifikation ermöglichen und Zertifikate darüber hinaus den Vorteil haben, zusätzlich bei beliebigen anderen Diensten ohne großen Aufwand eingesetzt werden zu können, wird eine zertifikatebasierte Authentifizierung bei der in dieser Arbeit entwickelten Lösung gewählt.

2.3 Protokolle

Unabhängig von der Übertragungstechnologie wickeln zwei miteinander Kommunizierende Endgeräte verschiedene Protokolle ab. Es gibt viele unterschiedliche Protokolle, die hier zum tragen kommen.
Das bekannteste und aufgrund seines weltweit in den meisten Netzen unabhängig von den Basistechnologien verwendete Protokoll ist das Internet Protokoll IP. Die neue Version 6 wurde bereits 1999 verabschiedet, konnte aber die alte Version noch nicht verdrängen. Der wesentliche Unterscheid zwischen dem alten IPv4 und neuen IPv6 besteht in der Vergrößerung des Adressraums. Während in Version 4 der Adressraum nur ~ 4,3 $*10^9$ verschiedene Adressen zulässt, ermöglicht IPv6 ~ 3,4$*10^{38}$ verschiedene Adressen. Eine Erweiterung von IP namens Mobile IP kurz MIP soll die Mobilität von Endgeräten, d.h. den Wechsel zwischen unterschiedlichen Netzen unterstützen. IP wurde mit einer Sicherheitserweiterung „Internet Protocol Security“ kurz IPSec genannt ergänzt, welche auf IP-Paketebene die Möglichkeit zur Authentifikation und Verschlüsselung bietet.
Oberhalb der IP-Ebene bietet das Secure Socket Layer (SSL) Protokoll bzw. sein Nachfolger das Transport Layer Security (TLS) Protokoll ebenso die Möglichkeit zur Authentifizierung zweier Endgeräte und Etablierung einer sicheren Verbindung durch Verschlüsselung.
Im Folgenden werden die beiden für sicheres Roaming wesentlichen Protokolle MIP und IPSec beschrieben und mit dem Protokoll SSL/TLS verglichen.

2.3.1 Mobile IP

Mobile IP (MIP) [Dixit02, Schiller03, Gho02, RFC-3344] ist eine Erweiterung des klassischen Netzwerkschichtprotokolls IP, das in den 60er Jahren für die Kommunikation zwischen nicht mobilen Computern entwickelt wurde [RFC-2002]. MIP wurde von der Internet Engineering Task Force kurz IETF 1996 publiziert. Dies trägt der in den letzten Jahrzehnten immer zunehmenden Zahl mobiler Endgeräte Rechnung, deren Anzahl auch weiterhin anwachsen wird. Die wachsende Anzahl an mobilen Computern wird mit Sicherheit auch immer mehr die Notwendigkeit für ein zuverlässiges Netzwerkprotokoll für mobile Endgeräte erforderlich machen. Dieser Entwicklung soll Mobile IP gerecht werden. Alle bisher verwendeten Netzwerkprotokolle gehen von einem an einem bestimmten und immer gleich bleibenden Punkt des Netzes angeschlossenen Endgerät - i. d. R. einem Computer - aus, das sich über eine Adresse als Mitglied ”seines“ Netzes identifiziert.
Der Grund für die Notwendigkeit der Erweiterung von IP besteht darin, dass ein mobiles Endgerät MH (Mobile Host), das sein Heimatnetz (HN) verlässt, um an einem anderen Netzzugangspunkt angeschlossen zu werden, vor der Wahl steht,

- entweder seine IP- Adresse zu behalten mit der Konsequenz, dass es von bereits gesendeten IP-Pakete nicht mehr erreicht wird und von ihm gesendete Pakete wegen einer topologisch nicht korrekten Absenderadresse ihr Ziel nicht mehr erreichen
- oder seine IP-Adresse zu ändern mit der Konsequenz, dass bestehende Verbindungen von der Schicht 4 an aufwärts abbrechen, wie z.B. TCP oder UDP

Verbindungen. Für sie wäre nach jedem Netzwechsel in diesem Fall ein Neuaufbau aller Verbindungen nötig.

Das Ziel von Mobile IP ist es, die Mobilität des Rechners für höhere Schichten transparent zu gestalten. MIP erweitert IP derart, die es einem MH ermöglicht, sich unter Beibehaltung seiner topologisch korrekten IP- Adresse frei zu bewegen. MIP definiert zu diesem Zweck die folgenden speziellen Objekte:

- einen Home Agent (HA),
- einen Foreign Agent (FA)
- sowie eine Care of Adresse (CoA).

Die Agenten sind Softwarekomponenten auf Rechnern, die in ihren jeweiligen Netzen Pakete abfangen und weiterleiten.

Der in Abbildung 9 schematisch dargestellte Nachrichtenaustausch bei der Ankunft eines MH an einem Netz bis zum möglichen Empfang von Daten ist folgender:
Durch Agent Advertisement oder Agent Solicitation machen sich der MH und je nach dem, ob sich der MH an seinem Heimatnetz oder an einem fremden Netz (FN) anmeldet, der entsprechende Mobilitätsagent HA oder FA gegenseitig bekannt. Erhält ein MH ein Agent Advertisement, kann er feststellen, ob er sich in seinem Heimatnetz oder in einem Fremdnetz befindet.
Befindet sich der MH in seinem Heimatnetz, wird keine Mobilitätsunterstützung benötigt. War der MH vorher in einem Fremdnetz (FN) registriert, so muss beim HA die dort registrierte CoA des Fremdnetzes gelöscht werden.
Befindet sich der MH in einem Fremdnetz, wird ihm vom Fremdnetz über das Advertisement eine CoA zugeteilt. Es gibt zwei Varianten, wie der MH zu einer CoA gelangen kann:
Die CoA ist identisch mit der Adresse des FA: Der Vorteil dieser Methode ist, dass mehrere MHs die gleiche CoA verwenden können. Der FA sorgt dann für die Zustellung der Pakete an die MHs, indem er die getunnelten Pakete auspackt und sie gemäß der Heimatadresse an den MH schickt. FA und MH sind in diesem Fall zwei separate Knoten.
Der MH erhält eine eigene sogenannte Colocated CoA: IP Adressen können an die MHs z.B. über Dynamic Host Configuration Protocol (DHCP) vergeben werden. Der MH ist sozusagen MH und FA in einem. Vorteil dieses Verfahrens ist, dass es auch dann funktioniert, wenn es in dem Netz keinen FA gibt. Der Nachteil ist, dass freie IP Adressen knapp werden könnten.

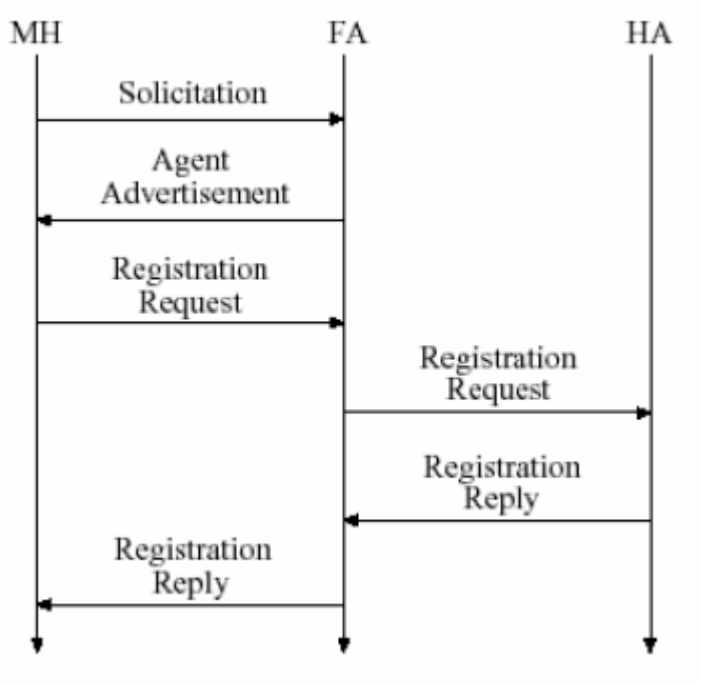


Abbildung 9: Nachrichtenaustausch bei Ankunft eines MH in einem FN

Die CoA muss dem HA über einen Registration Request mitgeteilt werden. Der HA antwortet daraufhin mit einem Registration Reply. Pakete, die an die Heimatadresse des MH geschickt werden, werden vom HA zur CoA getunnelt, ausgepackt und dem MH zugestellt. Hinter der CoA kann entweder der FA stehen oder im Falle einer Colocated CoA ist damit der MH selbst gemeint. Für das Versenden von Daten durch den MH kann Standard IP verwendet werden.

Den Kommunikationsprozess zwischen einem Correspondent Node (CN) und dem MH gemäß MIP wird in Abbildung 10 dargestellt. Man bezeichnet diesen Vorgang als Triangular Routing. Wenn ein CN eine Nachricht an den MH schicken will (#1), so muss er sie zuerst an den HA schicken, da nur er die aktuelle CoA des MH kennt. Der HA tunnelt die Pakete zur aktuellen CoA des MH (#2). Tunneln der Pakete vom HA zur CoA des MH ist notwendig, da die Zieladresse der Pakete, die an den MH gesendet werden die Heimatadresse des MH ist, wohin sie nach dem Standard Routing Protokoll im Internet gesendet werden. Mechanismen zum Tunneln sind mit IP-in-IP in [RFC-2003] und Generic Routing Encapsulation (GRE) in [RFC-2784] definiert. Der HA verpackt diese Pakete neu und versieht den neuen äußeren Header mit der CoA des MH. Die Pakete erreichen nun nach Standard Routing Protokollen das gewünschte Ziel.
Der FA entfernt den äußeren Header wieder - packt die Pakete also aus - und leitet sie an den MH weiter(#3). Der MH als Sender verschickt seine Pakete, direkt an den Empfänger (#4)+(#5).

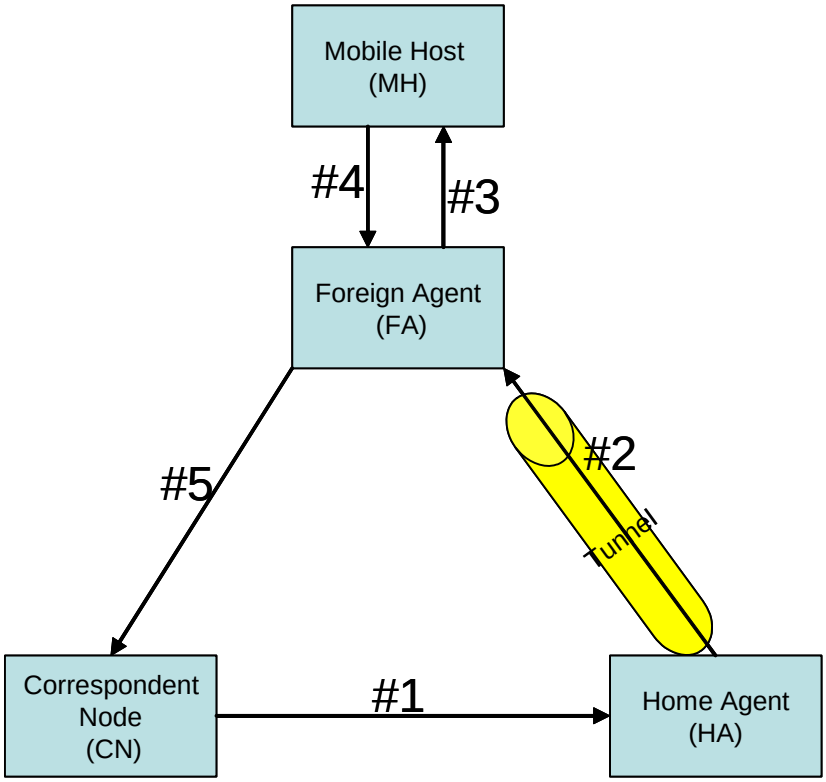


Abbildung 10: Triangular Routing

2.3.1.1 Vergleich von IPv4 und IPv6

Das im vorigen Abschnitt gesagte gilt für das Internet Protokoll der Version 4 (IPv4). Der Standard IP Version 6 (IPv6) weißt eine ganze Reihe von Vorteilen gegenüber der alten Version 4 auf. Im Moment sind die meisten Computernetze Mischnetze, die sowohl aus alten Komponenten, welche nur Version 4 beherrschen, als auch neue Komponenten, die schon über einen IPv6 Stack verfügen, bestehen. Die gängigen Betriebssysteme, wie Microsofts Windows XP, Apples OS-X und die meisten Linuxderivate verfügen im Moment über zwei IP-Stacks parallel - je einen für IPv4 und IPv6. Langfristig wird die neue Version IPv6 die alte Version 4 vollständig ablösen. Dies wird jedoch noch ein paar Jahre dauern.

Der IPv6 Header ist gegenüber dem IPv4 Header wegen der längeren Adressen auf von 20 Bytes bzw. mit Optionen 24 Bytes auf 40 Bytes angewachsen. Dies spielt bei den heute schon breitbandigen Netzen jedoch kaum eine Rolle. Da zukünftig die Bandbreite auch weiter anwachsen wird, fällt dies längerfristig nicht ins Gewicht. Bei schmalbandigen Verbindungen, wie sie ISDN oder GSM bieten, oder bei aus irgendwelchen Gründen sehr teuren Verbindungen kann man, falls die Protokoll-Header im Vergleich zu den Nutzdaten unverhältnismäßig lang sind, die Header komprimieren. Hierbei erreicht man bei IPv6 sogar eine geringfügig bessere Kompression als bei IPv4. Der Header von IPv4 ist in der folgenden Abbildung 11 dargestellt.

<table>
<tr><td>Version=4 (4 Bit)</td><td>HeaderLen (4 Bit)</td><td>Type Of Service (8 Bit)</td><td colspan="4">Paketlänge (16 Bit)</td></tr>
<tr><td colspan="3">Paket ID (16 Bit)</td><td>0</td><td>DF</td><td>MF</td><td>Fragment Offset (13 Bit)</td></tr>
<tr><td colspan="2">TTL (8 Bit)</td><td>Protokoll (8 Bit)</td><td colspan="4">Header-Prüfsumme (16 Bit)</td></tr>
<tr><td colspan="7">Quell-Adresse (32 Bit)</td></tr>
<tr><td colspan="7">Ziel Adresse (32 Bit)</td></tr>
<tr><td colspan="7">Optionen und optionales Padding (32 Bit)</td></tr>
</table>

Abbildung 11: IPv4 Header

Eines der zentralen Ziele der IPv6-Spezifikation ist die Vereinfachung der Header-Struktur. Im Vergleich zur obigen Abbildung 11 ist in Abbildung 12 die Struktur des IPv6 Header dargestellt. Die neue Header-Struktur soll sicherstellen, dass die erforderliche Flexibilität und Erweiterbarkeit für kommende Entwicklungen gewährleistet ist. Der IPv6-Header enthält neben der Versionsbezeichnung sieben Informationen:

- Priority: Erlaubt das Setzen von Prioritäten. In dem 4-Bit-langen Feld kann die Quelle einzelnen Paketen Prioritäten relativ zu anderen Paketen zuweisen.
- Flow Label: Definiert eine Art Verbindungs-ID zwischen zwei Endpunkten. Router können anhand dieser Informationen Pakete, die zu einer Verbindung gehören, direkt übermitteln, ohne die übrigen Header-Informationen zu analysieren. Diese Funktion ist vor allem für Multimedia-Anwendungen interessant.
- Payload Length: Gibt die Länge des IPv6-Datenpakets als Integer-Wert an. Ist das Paket größer als 64 KByte, wird der Wert auf 0 gesetzt und die exakte Länge im Options-Header angegeben.
- Next Header: Gibt den Header-Typ an, der auf den IPv6-Header folgt, beispielsweise Routing- oder Options-Header.
- Hop Limit: Bestimmt die maximale Anzahl an Routern, die ein IPv6-Datenpaket überqueren kann, ähnlich dem Time-to-Live-Wert bei IPv4.
- Source Address: Gibt die 128-Bit-lange IP-Adresse des Senders an.
- Destination Address: Gibt die 128-Bit-lange IP-Adresse des Empfängers an.

Version=6 (4 Bit)	Traffic Class / DS (8 Bit)	Flow Label (20 Bit)	
Länge der Nutzdaten (16 Bit)		Flags (8 Bit)	Hop Limit (8 Bit)
Quell-Adresse (128 Bit)			
Ziel-Adresse (128 Bit)			

Abbildung 12: IPv6 Header

2.3.1.2 Mobile IPv6

Die Architektur von Mobile IPv6 wird durch einige Neuerungen vereinfacht. Beim IPv6 gibt es keine FAs mehr, da in IPv6 die Verfügbarkeit von Adressen kein Problem darstellt. IPv6 erlaubt bis zu $2^{128} = 3.4028 \times 10^{38}$ adressierbare Knoten. Wenn man von einer Weltbevölkerung von 10^{10} Menschen ausginge, würden damit auf jeden Menschen mehr als 10^{28} Adressen fallen. Dies macht deutlich, dass der Adressraum für einen weltweiten Einsatz hinreichend groß ist. Ein weiterer Vorteil den IPv6 bietet ist die Möglichkeit der „Route Optimization“ (RO): Nachdem der MN sein aktuelles Binding an den CN geschickt hat, kann der CN ihn direkt über die CoA erreichen, wodurch ermöglicht werden soll, den besten Übertragungsweg zwischen CN und MN zu nutzen.

In Mobile IPv6 erhält ein MH jedes Mal, wenn er sich von einem Subnetz zu einem anderen Subnetz bewegt eine neue CoA, die immer colocated ist. Er registriert diese CoA bei seinem HA (#1, #2). Dieser arbeitet quasi als Proxy für den MH, bis dieser Binding Eintrag im Router ausläuft. Der HA fängt Pakete, die an die Heimatadresse des MH gerichtet sind, ab und leitet diese unter Benutzung der IPv6 Kapselung an die CoA des MH weiter (#3). Ebenso informiert der MH auch seine CHs per Binding Update (BU) über seine neue CoA (#5, #6). Wenn diese in der Lage sind, Änderungen in ihren lokalen Wegewahltabellen durchzuführen, schicken sie die Pakete dann direkt an den MH (#7), ansonsten muss der Weg über den HA gewählt werden (#3, #4). Bedenkt man, dass die Zahl der mobilen Hosts in Zukunft immer mehr ansteigen wird, führt dies dazu, dass die Zahl der BUs proportional dazu mit ansteigen und damit auch zu einer deutlichen Mehrbelastung von Netzressourcen führen wird.

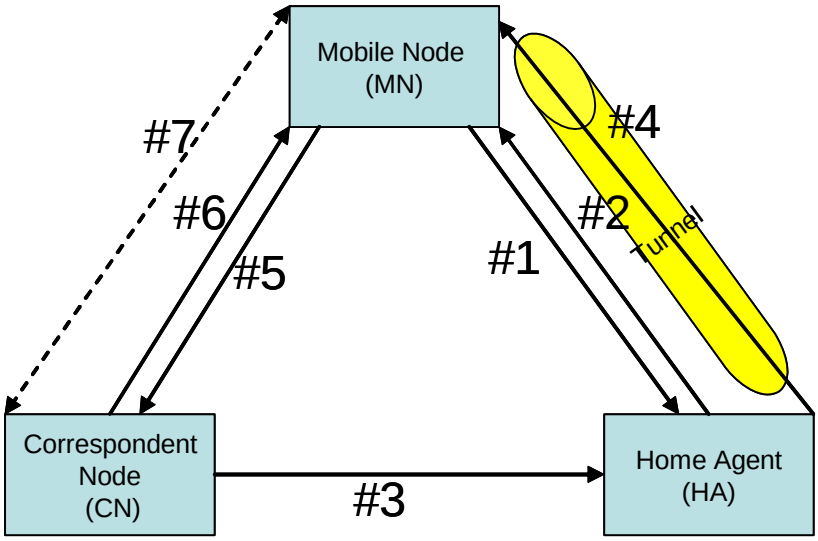


Abbildung 13: Direkte Kommunikation zwischen CN und MN in MIPv6

Alternativ zu der in der obigen Abbildung 13 gezeigten Kommunikation ist immer auch die Möglichkeit der indirekten Kommunikation über den HA gegeben. Die bedeutet, dass der CN alle Pakete an den HA sendet und der HA dann die Pakete zum MN tunnelt. Der MN seinerseits schickt jedoch die Pakete direkt zum CN.
In jedem Paket, welches vom MN zum CN geschickt wird, wird die so genannte „Home Option“ benutzt. Im äußeren IP Header ist dabei die CoA des MN als Absenderadresse eingetragen. Wenn ein Knoten, wie der CN ein Paket mit der „Home Option“ erhält, ersetzt er die im äußeren Header platzierte Absenderadresse durch die Heimatadresse bevor das Paket an die höheren Schichten weiter gegeben wird. Für die darüber liegenden Schichten scheint das Paket damit von der Heimatadresse zu kommen.

2.3.2 IPSec

IP-Pakete haben keinerlei inhärente Sicherheit. Die Adressen von Ix-Paketen können gefälscht, ihr Inhalt verändert und alte Pakete nochmals gesendet werden. IP Security (IPSec) [Dav01, Dixit02, NaDoHa00] bietet einen Schutz für IP-Datagramme. Es gibt zwei unterschiedliche Modi:

- den Transportmodus und
- den Tunnelmodus.

Im Transportmodus werden die Sicherheitsdienste der Protokolle nur auf die Pakete höherer Schichten angewandt, während sie sich im Tunnelmodus auf ganze IP-Pakete erstreckt. Im Transportmodus wird ein IPSec-Header zwischen dem IP-Header und dem Header des übergeordneten Protokolls eingefügt. Im Tunnel-Modus wird das ganze zu schützende IP-Paket in ein anderes IP-Datagramm eingekapselt, was in der folgenden Abbildung 14 deutlich wird:

Original IP-Paket	IP-Header	TCP-Header	Daten		
Geschütztes Paket im Transport-Modus	IP-Header	IPSec-Header	TCP-Header	Daten	
Geschütztes Paket im Tunnel-Modus	IP-Header	IPSec-Header	IP-Header	TCP-Header	Daten

Abbildung 14: Aufbau von IP-Paketen

Beide IPSec-Protokolle Authentication Header (AH) und Encapsulation Security Payload (ESP) können sowohl im Transport- als auch im Tunnel-Modus eingesetzt werden. In der Praxis wird AH im Tunnelmodus jedoch nicht benutzt, da dieselben Daten geschützt werden, wie bei AH im Transportmodus.

2.3.2.1 ESP

Encapsulation Security Payload (ESP) sorgt für Vertraulichkeit, Integrität, Authentifizierung der Datenquelle und optional Schutz vor Wiedereinspielung von Paketen [RFC-2406]. Um Vertraulichkeit und Authentifizierung zu gewährleisten stehen bei ESP verschieden Algorithmen zu Verfügung:

- Cipher für die Vertraulichkeit und
- ein Authentifikator für die Authentifizierung.

Es ist möglich eine Null-Cipher und einen Null-Authentifikator zu definieren und ESP entweder ohne Verschlüsselung oder ohne Authentifizierung zu betreiben. Es ist nicht möglich sowohl die Null-Cipher als auch den Null-Authentifikator gleichzeitig einzusetzen. Gemäß Spezifikation müssen als Authentifikator HMAC-MD5 und HMAC-SHA-1 vorhanden sein. Als Verschlüsselungsalgorithmen (Cipher) sind DES und 3DES gemäß RFC2406 auf jeden Fall vorzusehen. Des Weiteren müssen alle Verschlüsselungsalgorithmen bei ESP im CBC-Modus arbeiten. Abbildung 15 zeigt das Format eines mit ESP geschützten IP-Paketes:

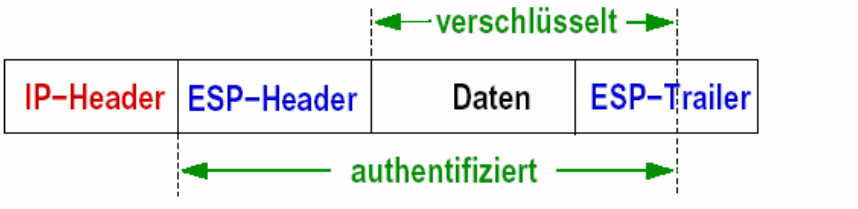


Abbildung 15: IP-Paket mit ESP Header [Eckert02]

Es werden ein Header und ein Trailer eingefügt. Der ESP-Header wird im Klartext übermittelt. Der Trailer wird zum Teil verschlüsselt. Das Format eines ESP-Paketes veranschaulicht folgende Abbildung 16:

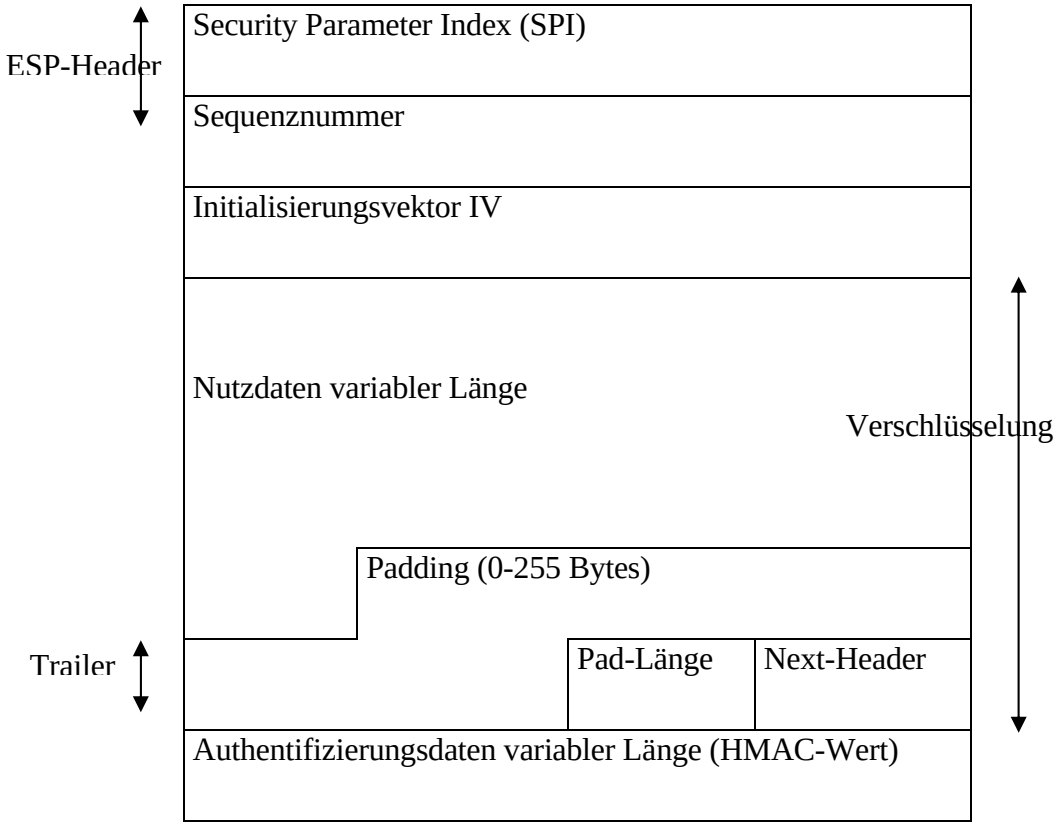


Abbildung 16: Format eines ESP-Paketes

2.3.2.2 AH

Der Authentication Header (AH) hat die Aufgabe die Quelle eines Datenpaketes zu authentifizieren, die Integrität des Paketes sicher zu stellen und optional Wiedereinspielungen zu erkennen und abzuwehren. Dazu wird vor die Daten ein AH-Header gehängt, so dass sich das in Abbildung 17 dargestellte Format eines IP-Paketes ergibt:

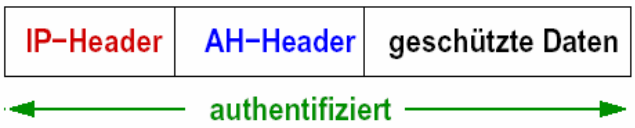


Abbildung 17: IP-Paket mit AH Header [Eckert02]

Das AH-Protokoll bietet seine Authentifikations- und Integritätsfunktionalität als verbindungslosen Dienst an. Daher kann ohne die optionalen Maßnahmen zur Abwehr von Replays, wie z.B. Sequenznummern, nur die Integrität einzelner Datenpakete nicht aber die Integrität der sich aus den Paketen zusammensetzenden Nachricht gewährleistet werden. Ein AH-Header ist, wie in Abbildung 18 dargestellt, aufgebaut.

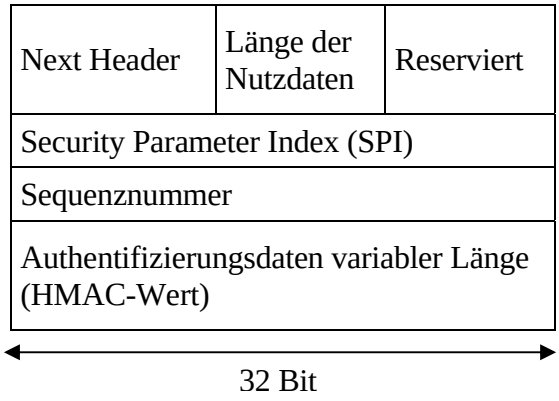


Abbildung 18: AH Header

Der reservierte Bereich ist für eine zukünftige Nutzung vorgesehen und muss den Wert Null enthalten. Der Absender eines Paketes gibt denjenigen SPI-Wert an, der die benötigte Security Association (SA) in der Datenbank des Empfängers identifiziert. Die Sequenznummern werden monoton erhöht. Um das Wiedereinspielen von Paketen zu verhindern, muss der Sender dafür sorgen, dass Sequenznummern sich nicht zyklisch wiederholen. Dazu etabliert er stets eine neue SA mit einem neuen MAC-Schlüssel, wenn die Sequenznummer ihren maximalen Wert erreicht hat. Gemäß Spezifikation [RFC-2402] ist vorgeschrieben, dass mindestens HMAC-MD5 und HMAC-SHA-1 anzubieten sind.
Wenn sowohl AH als auch ESP im Transportmodus benutzt werden, dann sollte ESP zuerst angewandt werden, da anderenfalls die Integrität nur für die Nutzdaten gegeben wäre. Wenn ESP zuerst angewandt wird, ist Integrität für die ESP-Nutzdaten gegeben, welche die zu transportierenden Nutzdaten enthalten.

2.3.2.3 Schlüsselmanagement

IPSec wird in Kombination mit einem Schlüssel-Management Protokoll eingesetzt. Als Schlüssel-Management Protokolle werden u. a. ISAKMP, Oakley und IKE eingesetzt. Diese werden im Folgenden beschrieben.

2.3.2.3.1 ISAKMP

Mit dem Internet Security Association and Key Management Protocol (ISAKMP) werden Protokolle und Formate spezifiziert, um SAs einzurichten, auszuhandeln, zu modifizieren und zu löschen. IKE ist ein Bestandteil von ISAKMP.

ISAKMP wird als separater Prozess im Kernel des Betriebssystems verankert und stellt anschließend für beliebige Instanzen (wie z.B. IPSec) Schlüsselmaterial bereit. Es verfolgt einen Zwei-Phasen-Ansatz:

In der ersten Phase wird ein sicherer Kanal zwischen zwei Key-Management-Instanzen zweier Computersysteme hergestellt und erhalten (asymmetrisch, meist mit Public Keys), über den dann in der zweiten Phase schnell und ohne Public-Key-Verfahren die symmetrischen Schlüssel ausgetauscht oder erneuert werden können.

ISAKMP-Pakete werden per UDP verbindungslos transportiert und jeder Nachricht geht ein ISAKMP-Header voraus. Teile dieses Headers (Cookies, Message ID) halten dabei einen minimalen Verbindungskontext, obwohl UDP ja ein nicht verbindungsorientiertes Protokoll ist, während die restlichen Header-Felder (Flags, Exchange Type, Length) für die Interpretation der nächsten Payload-Felder benötigt wird. Anschließend an den Header werden die Payload-Felder übertragen, die je nach Protokollschritt anders aussehen und auch verschlüsselt sein können. Dabei kommen folgende Schritte in Frage: SA-Etablierung, Proposal, Transform, Key Exchange, Certificate, CERT Request, Hash, Signature, Nonce (Nullfunktion), Notification, Delete.

In der ersten Phase wird mit starken kryptographischen Verfahren die Authentisierung des Kommunikationspartners vorgenommen, anschließend wird ein kryptographisches Verfahren zum Schutz der Verbindung ausgehandelt und dann werden die Schlüssel dafür etabliert. Eine eigene Security Association, die ISAKMP-SA, wird erstellt und die beiden Parteien stehen in einem Initiator – Responder Verhältnis zueinander. Das angewendete Verfahren ist sicher gegen Hijacking (Man-in-the-middle wird z.B. durch kryptographische Verknüpfung der Protokollnachrichten verhindert) und gegen DoS (wie z.B. TCP-SYN-Flooding), da in der Prä-Authentisierungsphase kaum ressourcenintensive Operationen durchgeführt werden und ebenfalls der Cookie- Mechanismus von Photuris zum Einsatz kommt [Mar00].
In der zweiten Phase ist ISAKMP in der Lage, jeder anfordernden Instanz (wie z.B. IPSec) einen sicheren Kanal zur Authentifizierung oder Schlüsselablieferung zur Verfügung stellen. Dank dieses sicheren Kanals kann rasch ein Key-Exchange erfolgen, geschützt durch die Security Services der Phase 1 (der ISAKMP-SA). Erfolgt während der zweiten Phase, z.B. bei der Übermittlung der symmetrischen Schlüssel, ein Fehler, muss nicht die ganze Verbindung neu aufgebaut werden – der Kanal aus der Phase eins steht immer noch zuverlässig zur Verfügung. ISAKMP verfügt über verschiedene Exchange-Types: den Base Exchange, den ID Protection Exchange, den Authentication only Exchange und den Aggressive Exchange. Jeder Typ verfügt über unterschiedliche Eigenschaften im Bezug auf die Anzahl der ausgetauschten Nachrichten zur Verbindungsetablierung, der Funktionalität oder dem Schutz der Identitäten

2.3.2.3.2 OAKLEY

Das Oakley Key Determination Protocol (Oakley) ist ein Austauschprotokoll für Diffie-Hellmann (DH) Schlüssel, das zwischen den Kommunikationspartnern Sicherheitsparameter aushandelt und festlegt. Eine wichtige Eigenschaft von Oakley ist

die Anti-Clogging-Defense gegen Denial of Service Attacks, die bereits in einer frühen Phase durch den Austausch von Cookies vor solchen Attacken weitgehend schützt. Dieses Cookie-System wurde aus Photuris übernommen, einem anderen hier nicht besprochenen Key Management Protokoll [KaSi99].
Zu Beginn der Verbindung werden die Verschlüsselungs-, Authentifizierungs- und Schlüsselgenerierungsmethoden etabliert und untereinander abgeglichen. Oakley ist außerdem in der Lage, die DH-Exponenten (also z.B. öffentliche Schlüssel) zu authentifizieren. Vor dieser Authentifizierung wird keine Exponentiation der DH-Exponenten durchgeführt – rechenintensive Operationen werden also erst gestartet, wenn die Identität des Kommunikationspartners gewiss ist, was ebenfalls DoS-Attacken erschwert. Für den Schlüssel kombiniert Oakley einen DH-Mechanismus mit einem Authentifizierungsverfahren (z.B. RSA). Im sogenannten Aggressive Mode braucht Oakley mindestens drei Nachrichten für die Etablierung der Verbindung, welche anschließend Perfect Forward Secrecy[15] bietet und die Identitäten geheim halten kann. Im Main Mode braucht Oakley mehr als drei Nachrichten für die Etablierung der Verbindung, bietet aber den vollen Schutz vor DoS-Angriffen durch den Austausch der Cookies, Perfect Forward Secrecy für die Geheimhaltung der Identitäten und Signaturen, damit beide Parteien im Nachhinein die Kommunikation sicher beweisen können.

2.3.2.3.3 IKE

Internet Key Exchange (IKE) [HC98] basiert auf den Nachrichten und Exchange Typen von ISAKMP. Es arbeitet mit dem Diffie-Hellmann-Public-Key-Verfahren, um einen authentischen Kanal zu erzeugen und einen Man-in-the-middle-attack auszuschliessen. Es werden Schlüssel mit kurzer Lifetime generiert, mit denen die Parameter signiert oder verschlüsselt werden, um die Endpunkte der Verbindung kryptographisch zu authentisieren. Für dieses asymmetrische Verschlüsselungs- und Signaturverfahren wird RSA oder DSS (Digital Signature Standard) eingesetzt, wobei die zum Einsatz kommenden öffentlichen Schlüssel zertifiziert sein sollten. Außerdem kann IKE auch mit HMAC symmetrisch authentisieren unter Verwendung von MD5 oder SHA-1. IKE gilt als sicher, ist schnell realisierbar, die authentisierenden Daten können aber, auf Grund des DH-Verfahrens, nicht von Dritten geprüft werden. Nach der Authentisierung wird der Kanal mit einer symmetrischen Verschlüsselung geschützt (DES in CBC Mode gilt als Minimum).
IKE kennt verschiedene Phasen und Modi. Es verfolgt wie ISAKMP einen Zwei-Phasen-Ansatz: Die Etablierung eines sicheren Kanals als Phase 1, den Austausch weiterer Schlüssel über den sicheren Kanal als Phase 2. In der Phase 1 benutzt IKE dieselben Standard-Exchange-Typen wie ISAKMP, wobei je nach Modus 3 – 7 Nachrichten für die Kanaletablierung nötig sind.
Zwei Modi werden in der Phase 1 verwendet: Im Aggressive Mode tauscht IKE bloß drei Nachrichten für die Authentisierung aus, wobei dafür einige Nachteile in Kauf zu nehmen sind. Schon nach der ersten empfangenen Nachricht werden aufwändige Rechenoperationen nötig, die Verbindung ist weniger DoS-resistent, da kein Cookie-

[15] Bei Perfect Forward Secrecy (PFS) werden die einem abgeleiteten Schlüssel vorausgehenden und nachfolgenden Schlüssel nicht aus den vorherigen abgeleitet.

Exchange stattfindet und die SA-Parameter können nicht angeglichen werden. Die Identitäten der Kommunikationspartner können nur unter Verwendung von Public Key Encryption geschützt werden.
Im Main Mode werden zwar mehr Roundtrips benötigt, aber dafür auch einige Vorteile erreicht: Die DoS-Resistenz ist dank dem Cookie-Austausch höher und insbesondere die SA-Parameter sind angleichbar, so dass eine Art „Vertrag“ über die Sicherheitsbedingungen ausgehandelt werden kann. Die Authentisierung in dieser Phase erfolgt durch ein DH-Verfahren mit Zufallswerten, wobei drei verschiedene Verfahren angewendet werden können: Eine Signatur (DSS oder RSA; garantiert Nichtabstreitbarkeit der Kommunikationsbeziehung), Public Key Encryption (RSA; bietet keine Nichtabstreitbarkeit) oder Pre Shared Keys (keine Zertifikate notwendig). Die Verfahren sind nicht kombinierbar, allerdings ist der Algorithmus zur Generierung des endgültigen, symmetrischen Schlüsselmaterials der IKE-SA abhängig vom gewählten Authentisierungsverfahren. Die Payload der Pakete kann verschlüsselt werden, sobald beide Seiten die Parameter zur Generierung des Schlüssels haben. Im Aggressive Mode ist dies in der dritten, im Main Mode ab der fünften Nachricht möglich. Die Identitäten können im Main Mode geschützt werden, da sie erst in Nachricht vier übertragen werden, die bereits verschlüsselt ist. Im Aggressive Mode ist ein Schutz der Identitäten nur unter Verwendung von Public Key Exchange möglich. Die IP-Adresse bleibt aber aus dem UDP-Paket bestimmbar. Die Authentisierung und der Schlüsselaustausch werden strikte voneinander getrennt. Nach der Phase 1 stehen geheime Schlüssel und vereinbarte kryptographische Verfahren in der IKE-SA zur Verfügung. In der Phase 2 wird wie bei ISAKMP der sichere Kanal aus Phase 1 für den Austausch weiterer Schlüssel und die Etablierung weiterer SAs (z.B. für IPSec) verwendet. Durch den Austausch weiterer D-Exponenten über den sicheren Kanal kann beispielsweise Perfekt Forward Secrecy erreicht werden. Die Phase 2 benutzt ein schnelleres Protokoll als die Phase 1 und kennt auch nur einen Modus, den Quick Mode: Es werden nur drei Nachrichten ausgetauscht. Alle IKE-Client, wie beispielsweise IPSec, können auf der Basis von IKE schnell neue symmetrische Schlüssel generieren und sicher austauschen.

Die Abbildung 19 gibt vergröbert den Ablauf des IKE-Protokolls wieder:

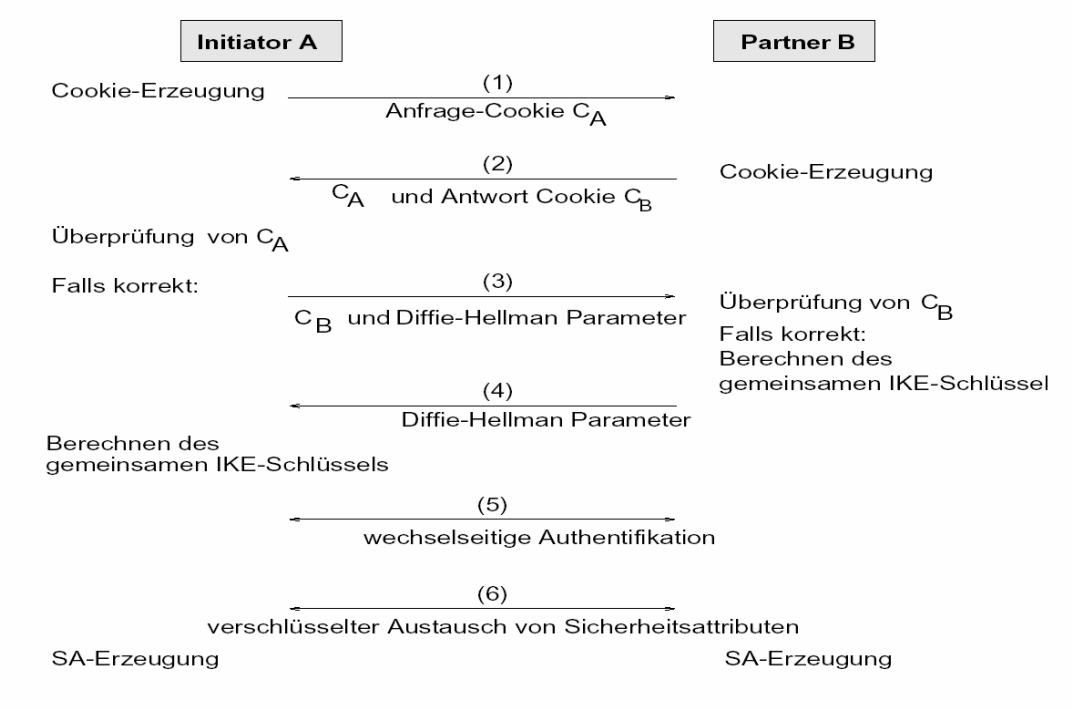


Abbildung 19: Ablauf des IKE Protokolls [Eckert02]

2.3.2.4 Unterschiedliche Sicherheitsstufen durch IPSec

IPSec bietet die Möglichkeit verschiedene Verschlüsselungsverfahren einzusetzen. So könnte man je nach Anspruch AES mit einem 256 bit langem Schlüssel, RC4 mit 128 bit langem Schlüssel oder auch gar keine Verschlüsselung einsetzen. Darüber hinaus können beide Kommunikationsrichtungen getrennt verschlüsselt werden, d.h. für die Hin und Rückrichtung können unterschiedliche Verfahren angewandt werden. Dies bedeutet, dass verschiedene Stufen der Sicherheit gewährleistet werden können je nach gewähltem Verfahren und Schlüssellänge.
Bei einem verhältnismäßig leistungsschwachen mobilen Endgerät könnte man z.B. die Kommunikationsverbindung mit dem hierfür geeigneten RC4 Verfahren [Schneier96] unter Verwendung eines 128 Bit langen Schlüssels verschlüsseln. Bei Einsatz eines Leistungsstarken Endgerätes können z. B. AES oder 3DES mit längeren Schlüsseln eingesetzt werden.

Ein wichitger Aspekt bei der Festlegung eines notwendigen Sicherheitsniveaus, sind auch sogenannte „Lawful-interception"-Interfaces, die das Mithören von Kommunikation durch Staatsorgane ermöglichen. Der ETSI Standard beschreibt ein Handover Interface für “Lawful Interception” zwischen Netzbetreibern, Dienstanbietern und „Law Enforcement Agencies“: Das Interface ES 201 671: "Handover Interface for the lawful interception of telecommunications traffic".
Es gibt Gerüchte, die besagen, dass derartige Schnittstellen in manchen Ländern nicht nur zur Verbrechensbekämpfung, sondern auch zur Erlangung wirtschaftlich relevanter Informationen genutzt werden.

In einem Unternehmen, in dem diesen Gerüchten glauben geschenkt wird, sollte daher grundsätzlich die Kommunikation zwischen einem Unternehmensmitarbeiter und dem Firmennetz mit AES oder 3DES bei Einsatz eines mindestens 256 bit langen Schlüssels verschlüsselt werden, unabhängig davon, ob das Zugangsnetz eigentlich schon eine sichere Verschlüsselung bietet oder nicht. Dem tragen die in Kapitel 3 beschriebenen VPN-Zugangsdienst Modelle Rechnung.
Da der Aufbau einer verschlüsselten IPSec Verbindung bei Einsatz des RC4 Verfahrens sich von dem einer mit AES verschlüsselten IPSec Verbindung nicht wesentlich unterscheidet, wird in den folgenden Kapiteln darauf nicht weiter eingegangen. Der Unterschied besteht nur darin, dass bei den Endpunkten die anderen Verfahren eingesetzt werden müssen und dass der eingesetzte Schlüssel die passende Länge haben muss. Des Weiteren wäre denkbar, dass die Partei, die die Etablierung einer solchen Verbindung als Dienst anbietet unterschiedlich hohe Gebühren für die unterschiedlich sicheren Verbindungen verlangt.

Wenn das Sicherheitsniveau im Rahmen einer IPSec-Verbindung in Abhängigkeit von den Leistungen des Zugangsnetzes gesetzt werden soll, ist ein Endgerät erforderlich, welches im Rahmen des in Kapitel 4 ausführlich beschriebenen Authentifizierungsvorganges mitteilt, über was für eine Art von Zugangsnetz es kommuniziert. Dies kann dem Prinzip eines „Quarantine Network“ (QN) folgend geschehen. Wenn ein Endgerät in ein solches Netz will, wird, bevor es Zugang erhält, zunächst geprüft, was für Versionen von diversen Applikationen oder auch des Betriebssystems vorhanden sind und gegebenenfalls werden „updates“ oder „patches“ eingespielt, bevor Zugang zum Netz gewährt wird oder der Zugang wird verweigert. Das Endgerät schickt im Rahmen der Authentifzierung signierte Informationen über sich selbst, d.h. die Art des Endgerätes, und die Art des Zugangsnetzes, über die es kommuniziert, und in Abhängigkeit einer festgelegten Strategie sorgt der in Kapitel 3 beschriebene Roaming Service Provider (RSP) dafür, dass der festgelegten Strategie folgend die notwendigen Maßnahmen, wie der Aufbau eines VPN, eingeleitet werden und dass nur auf bestimmte Ressourcen zugegriffen werden kann, so wie die entsprechende Strategie es für das Endgerät bei gegebenem Zugangsnetz vorsieht.

2.3.3 SSL/TLS

Das SSL Protokoll [SSL, SSL3] bietet ebenfalls die Möglichkeit eine sichere Verbindung zwischen zwei Endpunkten aufzubauen. Es wurde dafür ausgelegt eine sichere Verbindung zwischen einem Client und einem Server über einen unsicheren Kanal einzurichten. Als es 1999 von der Internet Engineering Task Force (IETF) als Standard festgelegt wurde, benannte man es in Transport Layer Security (TLS) Protokoll [DieAll99] um. SSLv3.1 ist identisch mit TLSv1.0. Die Unterschiede zwischen SSLv3.0 und TLSv1.0 sind marginal [BSI03]:

- TLS verwendet einen Hashed Message Authentication Code (HMAC) statt des MAC-Algorithmus von SSL 3.0.
- In TLS fiel die in SSL 3.0 hinzugekommene Unterstützung für die in den so genannten „Fortezza“-Karten in Hardware realisierten US-Behördenalgorithmen wieder weg.

- In TLS werden folgende zusätzliche Cipher Suites vorgeschlagen:
  - Elliptic Curve Cryptosystem (ECC) und
  - Kerberos 5.
- TLS schlägt als obligatorische Cipher Suite TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA vor[16]

SSL unterstützt drei Authentifizierungsarten:

- Die Authentifizierung beider Parteien,
- Die Authentifizierung des Servers gegenüber einem authentifizierten Client und
- Vollständige Anonymität.

Vollständig anonyme Sitzungen sind immer gegen Man-in-the-middle-Angriffe verwundbar. Im Rahmen dieser Arbeit geht es ausschließlich um eine beidseitige Authentifizierung. Jede Partei ist dabei für die Überprüfung des Zertifikates der anderen Partei verantwortlich und achtet darauf, dass es gültig und nicht abgelaufen oder gesperrt ist.

Im Einzelnen läuft das in Abbildung 20 dargestellte SSL-Handshake Protokoll zwischen einem Mobilen Endgerät bzw. dem Mobile Node (MN) und einem Authentifikations-Server folgendermaßen ab:

1. Der MN schickt ein MN-Hello–Paket an den Server. Dieses enthält unter anderem die Session-ID. Möchte der MN eine in der Vergangenheit ausgehandelte Verbindung wiederaufnehmen, dann wird die entsprechende Sitzungs-ID angegeben. Für die Einrichtung einer neuen Verbindung ist die ID Null. Des Weiteren enthält das Paket eine Liste mit den vom MN unterstützten Verschlüsselungsalgorithmen. Die im Paket enthaltene Zufallszahl findet später bei der Erzeugung des MasterSecrets Verwendung.

2. Der Server antwortet mit einem Server-Hello. Ist die in 1. vom MN empfangene SitzungsID ungleich Null, entscheidet der Server, ob er die alte Verbindung - vorausgesetzt den Fall es gab sie - wiederaufgenommen werden soll oder nicht. Wenn die alte Verbindung wieder aufgenommen werden soll, schickt er die gleiche SitzungsID zurück und der Handshake ist schon beendet. Soll eine neue Verbindung aufgebaut werden, wählt der Server aus der ihm zu Verfügung gestellten Liste je ein symmetrisches und asymmetrisches Verschlüsselungsverfahren aus. Auch „Server-Hello“ enthält eine Zufallszahl die später bei der Erzeugung des MasterSecrets Verwendung findet. Optional enthält das erste vom Server gesendete Paket die Anforderung eines Zertifikates an den MN. Mit „Certificate“ schickt der Server an den MN ein Zertifikat, welches seine Zugehörigkeit zu der seines im Zertifikat enthaltenen öffentlichen Schlüssels bestätigt. „Certificate“ ist optional. Der öffentliche Schlüssel kann auch

[16] Das bedeutet: Schlüsselaustausch mittels des Diffie-Hellman-Verfahrens, wobei die zum Schlüsselaustausch verwendeten Daten per DSS-Signatur authentisiert werden, Datenverschlüsselung mit Triple-DES im CBC-Modus (Schlüssellänge 168 Bit) und Integritätscheck mit HMAC-SHA-1

unzertifiziert mit „Server-Key-Exchange“ übermittelt werden, wenn kein Zertifikat vorliegt oder der MN nicht die Möglichkeit hat dieses zu überprüfen. Es muss auf jeden Fall sichergestellt werden, dass der MN den öffentlichen Schlüssel des Servers und keinen anderen erhält. Wenn man davon ausgeht, dass dieser Schlüssel schon auf dem MN vorhanden ist, nachdem er z.B. persönlich auf sicherem Wege dort installiert wurde, kann dieser Teil des Paketes natürlich entfallen. „Hello Done“ signalisiert das Ende des Server-Hello-Paketes.

3. Falls in 2. eine Zertifizierung des Servers angefordert wurde, enthält das zweite vom MN gesendete Paket mit „Certificate“ ein Zertifikat, welches seine digitale Unterschrift zertifiziert. Darin ist also der zur Überprüfung der digitalen Signatur nötige öffentliche Schlüssel des MN enthalten und es wird seine Zugehörigkeit zum MN bestätigt. handelt sich üblicherweise um ein X.509 – Zertifikat. Der „Certificate-Verify“-Teil des Paketes besteht aus dem mit dem zur MN-Signatur gehörigen privaten Schlüssel verschlüsselten Zertifikateteil – dem signierten Zertifikat also. Des Weiteren erzeugt der MN eine Zufallszahl das sogenannte PreMasterSecret. Er verschlüsselt es mit dem öffentlichen Schlüssel des Servers und sendet dieses als „Client-Key-Exchange“ in diesem Paket mit. „Change-Cipher-Spec“ beinhaltet lediglich die Mitteilung, dass ab jetzt verschlüsselt gesendet wird, ist selbst jedoch noch unverschlüsselt. Es wird nun aus dem PreMasterSecret und den beiden zu Beginn ausgetauschten Zufallszahlen Client.Zufallszahl und Server.Zufallszahl das MasterSecret berechnet. Gemäß SSL3.0 Spezifikation [SSL3] ist das MasterSecret 48 Byte groß und berechnet sich wie folgt:

```
MasterSecret= MD5( PreMasterSecret + SHA( ‚A’ + PreMasterSecret +
                   ClientHello.Zufallszahl + ServerHello.Zufallszahl ) ) +
                   MD5( PreMasterSecret + SHA( ‚BB’ + PreMasterSecret +
                   ClientHello.Zufallszahl + ServerHello.Zufallszahl ) ) +
                   MD5( PreMasterSecret + SHA( ‚CCC’ + PreMasterSecret +
                   ClientHello.Zufallszahl + ServerHello.Zufallszahl ) )
```

Aus dem MasterSecret wird schließlich der Schlüssel K erzeugt. Dies geschieht auf ähnliche Weise wie die Erzeugung des MasterSecrets aus dem PreMasterSecret. Im Wesentlichen also durch die Anwendung geeigneter Hashfunktionen. Die genauen Verfahren für die Erzeugung von Schlüsseln für die unterschiedlichen Verschlüsselungsverfahren finden sich in der SSL–Spezifikation. [SSL3]. K wird fortan für die Verschlüsselung aller Daten mittels des vereinbarten symmetrischen Verschlüsselungsverfahrens verwendet. Am Schluss des Paketes befindet sich „Finished“.

„Finished“ enthält einen Hashwert des MasterSecrets und aller bisher im Handshake ausgetauschten Nachrichten. „Finished“ ist mit dem vereinbarten Schlüssel K verschlüsselt.

4. Schließlich sendet auch der Server analog zum MN in dem den Handshake abschließenden Paket eine „ChangeCipherSpec und eine „Finished“ Mitteilung. Auch hierbei ist „ChangeCipherSpec“ noch unverschlüsselt. „Finished“ ist verschlüsselt.

Der weitere Datenaustausch erfolgt nun verschlüsselt mit den ausgehandelten Methoden.

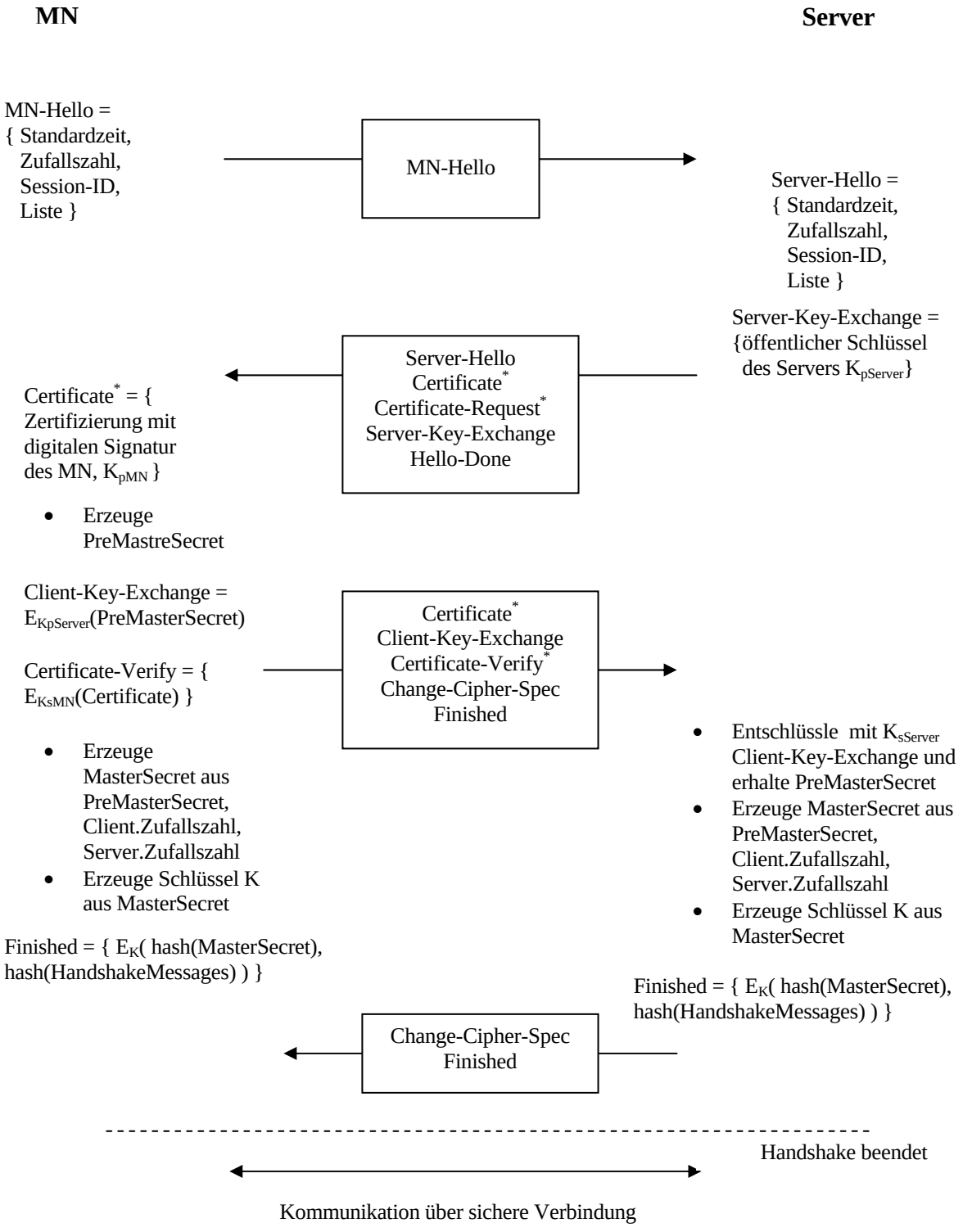


Abbildung 20: SSL/TLS Handshake Protokoll gemäß [Haisch01]

Da die im Rahmen dieser Arbeit implementierte Authentifikationslösung im Wesentlichen auf SSL/TLS beruht, wird im Folgenden Abschnitt noch besonders auf die Sicherheit des SSL Protokolls eingegangen.

2.3.3.1 Sicherheit des Handshake Protokolls

Formal wird die Sicherheit von SSL in [MiShSt98] und von TLS in [Paulson97] dargestellt. Die Sicherheit der eingesetzten Verschlüsselungsverfahren ist dabei immer Voraussetzung für die Sicherheit des Protokolls. Als unsicher angesehene Verfahren müssen durch neue sichere ersetzt werden, wenn Sicherheit gewährleistet werden soll. Es wird hier davon ausgegangen, dass die eingesetzten Verfahren sicher sind.

Es wird hier weiter davon ausgegangen, dass ein Angreifer alle übertragenen Datenpakete sieht. Für einen Angreifer ergeben sich dann folgende Möglichkeiten:

- Löschen von Datenpaketen
- Vervielfältigen von Datenpaketen
- Verändern von Datenpaketen

Das „Client-Hello“ ist wie auch das „Server-Hello“ unverschlüsselt. Es kann also beliebig abgehört und gefälscht werden. Dies ist jedoch dann nicht problematisch, wenn der MN sich sicher sein kann, den dem Server zugeordneten öffentlichen Schlüssel zu kennen. Da das PreMasterSecret mit dem öffentlichen Schlüssel des Servers verschlüsselt wurde, kann nur der Server es entschlüsseln. Wenn eine Zuordnung des Servers zu seinem öffentlichen Schlüssel nicht möglich ist, kann auch keine Sicherheit gewährleistet werden, da ein Angreifer seinen Schlüssel anstelle des Schlüssels des Servers schicken könnte. Die Zugehörigkeit des öffentlichen Schlüssel und des Servers sollte also zertifiziert sein.

Die ChangeCipherSpec Nachricht enthält keine Informationen, die ein Angreifer für sich nutzen kann. Wichtig ist wiederum dabei, dass beide Seiten auf ein korrektes „Finish“ warten, da es mit dem korrekten Schlüssel verschlüsselt ist. Um eine korrektes „Finish“ zu erzeugen, benötigt der Server das PreMasterSecret. Das bedeutet, dass ein Angreifer diese Nachricht nicht erzeugen kann, wenn man sowohl von einem sicheren symmetrischen Verschlüsselungsverfahren als auch von der Kenntnis des korrekten öffentlichen Schlüssels des Servers beim MN ausgeht.

Finish kann, wenn man von einem sicheren symmetrischen Verschlüsselungsverfahren ausgeht, auch nicht durch einen Angreifer gezielt inhaltlich verändert werden. Da es auch einen Hashwert aller vorangegangenen Nachrichten enthält, wird spätestens hier die Manipulation vorangegangener Nachrichten durch einen Angreifer deutlich.

Nach erfolgreichem Handshake kann der Angreifer Pakete nicht mehr gezielt verändern, da er ihren Inhalt aufgrund der Verschlüsselung ja nicht kennt. Eine willkürliche Manipulation kann durch hinzufügen eines Hashwertes der übertragenen Daten an ihren Schluss - vor der Verschlüsselung natürlich - jedoch leicht erkannt werden. Der Angreifer kann Pakete jedoch noch löschen oder vervielfältigen. Dies kann durch einfaches

Nummerieren der Pakete bemerkt werden. Da die Nummerierung ja mit den Daten verschlüsselt wird, kann sie auch nicht entdeckt und manipuliert werden.

In [Wagner96] wird das SSL Protokoll analysiert. Hierbei ist ein möglicher Angriff aufgeführt, der einen Spezialfall betrifft. Dies ist der Fall einer Ciphersuite, welche nur Nachrichtenauthentifizierung aber keine Verschlüsselung enthält. Die „change cipher spec“ Nachricht kann von einem Angreifer gelöscht werden, wodurch die gegenwärtige Ciphersuite beibehalten und nicht aktualisiert wird. In [Wagner97] wird bereits gesagt, dass dieser Fehler behoben ist. Die bei [Wagner97] aufgeführte Key-Exchange-Rollback Attacke lässt sich dadurch beheben, dass die Implemetierung prüft, dass die Anzahl der Felder der ServerKeyExchange Nachricht zur gewählten „Cipher suite“ passt. Ein weiteres wesentliches bei [Wagner96] aufgeführtes Problem ist das Problem, dass bei Einsatz von SSL3.0 aus Kompatibilitätsgründen in vielen Implementierungen die Möglichkeit eines Versionsrückfalls auf das bekannt unsichere SSL2.0 [Wagner97, ViMeCh02] besteht. Dies sollte auf keinen Fall möglich sein und ist bei den in dieser Arbeit vorgenommenen in Kapitel 9 beschriebenen Implementierungen auch nicht möglich.

Bei den in [Wagner96, Wagner97] aufgeführten „Schwächen“ handelt es sich nicht um grundlegende Schwächen des SSL Protokolls an sich, sondern um Fehler, welche durch schlechte Implementierung entstehen können: „these are not universal weaknesses: different implementations may or may not be vulnerable.“ Bei aktuellen SSL/TLS Implementierungen, wie dem hier für die Implementierung verwendeten OpenSSL, sollte dies berücksichtigt sein.

Gemäß [Eckert04] ergibt sich ein Sicherheitsproblem beim Einsatz von SSL im Zusammenhang mit Spoofing. Einem Angreifer ist es demnach möglich dem Nutzer gefälschte Seiten und damit auch gefälschte Zertifikatsinformationen vorzugaukeln. Das Konzept der Zertifikatsüberprüfung wird gemäß [Eckert04] weiterhin dadurch ausgehöhlt, dass bei den heutigen Systemen die bei der Installation vorgenommenen Voreinstellungen standardmäßig eine Vielzahl von Zertifizierungsinstanzen eingetragen sind, deren Zertifikaten automatisch vertraut wird ohne dass der Benutzer selber die Vertrauenswürdigkeit überprüfen kann. Diese Probleme sind bei der in dieser Arbeit entwickelten in Kapitel 4 dargestellten Architektur nicht gegeben, da zum einen kein Browser verwendet wird, der über derartige Voreinstellungen verfügt, sonder eine eigener SSL Client außerhalb und unabhängig von einem Browser und zum anderen wird die Zertifikateüberprüfung für beide Seiten Client und Server – also auch für den Nutzer - von einem vertrauenswürdigen PKI Server durchgeführt, wie in Kapitel 9 ausführlich dargestellt.

2.3.3.2 Vergleich von SSL mit IPSec

Folgende Tabelle 4 gibt zum Abschluss dieses Abschnitts noch einen Vergleich von IPSec mit dem populären Secure Socket Layer (SSL) Protokoll.

Tabelle 4: Vergleich SSL und IPSec

SSL	IPSec
Sichert nur TCP-basierte Anwendungen	Sichert IP-basierte Anwendungen
Pro Sitzung/Verbindung: dynamische Vereinbarung der Verfahren	Policy-DB+ SA-Festlegung rel. statisch, effizient unterschiedliche Verfahren , Protokolle für Hin- u. Rückrichtung möglich
Sehr einfach, automatischer Einsatz der Verfahren für beide Richtungen	flexibel aber komplex
Keine Verschleierung der Verkehrsdaten	Tunnel-Mode zum Verschleiern der Verkehrsdaten

2.4 Roaming

Übergaben zwischen zwei Funkzellen eines Funkdienstes werden als horizontaler Handover bezeichnet. Wechselt man zu einem anderen Funkdienst spricht man von einem vertikalen Handover. Ein übergangsloser oder „seamless“ Handover beinhaltet zudem noch die Aussage, dass eine begonnene Sitzung auch nach dem Wechsel von einem Funknetz ins andere erhalten bleibt. Als Roaming wird eine besondere Art des Handovers bezeichnet. Roaming beinhaltet den Wechsel zwischen Netzen unterschiedlicher Betreiber.

Die beiden wichtigsten Arten von Zugangsnetzen sind zum jetzigen Zeitpunkt WLAN und UMTS als Nachfolger von GSM. Die Kombination der beiden Technologien eröffnet prinzipiell für den Endnutzer die Möglichkeit, weltweit umherzuwandern und dabei permanent eine Verbindung zum Internet zu haben. Um dies zu verwirklichen würde zwar eine flächendeckende UMTS – Infrastruktur alleine schon ausreichen, jedoch die Möglichkeit zwischen UMTS und WLAN wechseln zu können, ist hierbei besonders interessant, um immer wenn möglich die höhere Bandbreite des kostengünstigeren WLANs nutzen zu können. Eine weltumspannende Infrastruktur, die einem Nutzer weltweit lückenlos ermöglicht „always on“ zu sein - idealerweise auch noch mit einem hohen Anteil von WLAN Nutzung - ist im Moment jedoch nicht gegeben. Es gibt lediglich lokale Insellösungen.

Die Standardisierungsgremien haben bereits Standards für die Zusammenarbeit der beiden Systeme definiert, wie z. B. im technischen Report Nummer 23.934 Release 6 von 3GPP [23.934]. Die „Telecoms & Internet converged Services & Protocols for Advanced Network“ (TISPAN) Kompetenzgruppe der ETSI [www.etsi.org/tispan/] arbeitet darüber hinaus mit 3GPP zusammen an einer IMS basierten Architektur zur Harmonisierung von festen und drahtlosen Netzen.
Beim Übergang zwischen WLAN und UMTS sind im Wesentlichen zwei Lösungen vorgeschlagen worden. Die eine wird als „loose coupling“ und die andere als „tight coupling“ bezeichnet. Die folgende Abbildung 21 zeigt die beiden Ansätze.

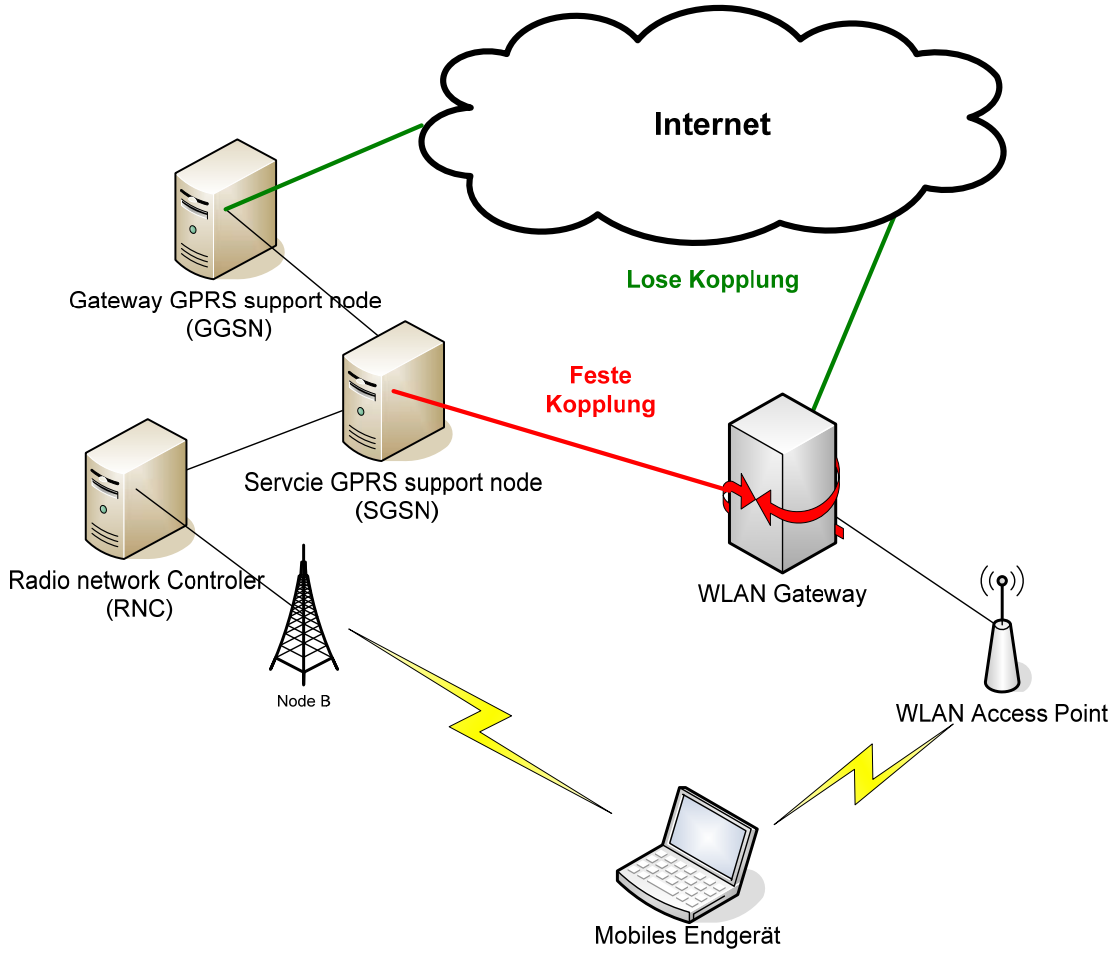


Abbildung 21: Lose und feste Kopplung zwischen WLAN und UMTS

2.4.1 Fest verkoppelte Zusammenarbeit

Hier ist das WLAN direkt mit dem UMTS Kernnetz (Core network) verbunden. Bei dieser Lösung müssen GGSN und SGSN den höheren möglichen Datenraten des WLAN angepasst werden. Auf der Seite des WLAN muss ein entsprechend modifiziertes WLAN Gateway vorhanden sein, das es ermöglicht die Information aus dem WLAN direkt in das UMTS Kernnetz zu übertragen. Dieser Ansatz ist nur dann sinnvoll, wenn WLAN Betreiber und UMTS Netzbetreiber ein und derselbe sind. Darüber hinaus machen die Kosten für die Anpassung des Kernnetzes und des WLAN Gateways diesen Ansatz vergleichsweise ungünstig.

2.4.2 Lose gekoppelte Zusammenarbeit

Bei der losen Kopplung wird das WLAN als paketorientiertes Zugangsnetz in Ergänzung zum UMTS Netz gesehen. WLAN und UMTS sind getrennte Netze, die unabhängig voneinander exisitieren können. Es ist möglich, vom WLAN aus direkt über das WLAN Gateway ohne über den SGSN Knoten durch das UMTS Kernnetz zu müssen ins Internet

zu gelangen. Zwischen den Betreibern der beiden Netze können Vereinbarungen getroffen werden, um Roaming für die Endnutzer zu ermöglichen. Die notwendigen Modifikationen für das UMTS Netz sind bei dieser Variante vergleichsweise gering. Die im Rahmen dieser Arbeit entwickelte Sicherheitsarchitektur geht von einer losen Kopplung aus.

2.4.3 Interworking Architektur von 3GPP

In diesem Abschnitt wird zunächst die von 3GPP definierte Interworkingarchitektur beschrieben [3GPP23.234]. Sie ist in der folgenden Abbildung 22 dargestellt.

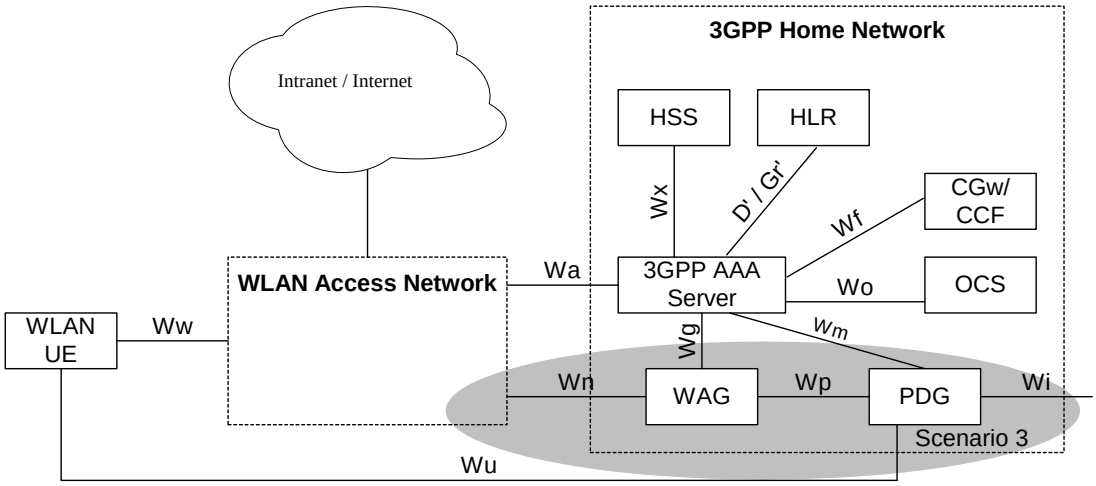


Abbildung 22: 3GPP Interworking Architektur [3GPP 23.234]

Die folgende List beschreibt die beim Zugang relevanten Komponenten, die in der obigen Abbildung dargestellt und die im Rahmen der Zugangskontrolle relevant sind, gemäß [3GPP23.234]:

- Das Nutzergerät **UE** (eventuell mit einer UICC Karte ausgerüstet) wird von einem 3GPP Teilnehmer dazu benutzt, auf den WLAN Interworking Dienst zuzugreifen. Das UE kann dabei nur fähig sein, auf das WLAN zuzugreifen oder es kann die Fähigkeit haben, sowohl auf das WLAN als auch auf das 3GPP System zuzugreifen. Einige UE können sogar die Fähigkeit haben, gleichzeitig auf beide, d.h. auf WLAN und auf 3GPP Systeme, zuzugreifen. Das UE kann Terminaltypen einschließen, deren Konfiguration (z.B. Interface zu einer UICC Karte), Operations- und Softwareumgebung nicht unter der alleinigen Kontrolle des Betreibers des 3GPP Systems sind. Z.B. kann ein UE ein Laptop oder ein PDA sein mit einer WLAN Karte, mit einem UICC Kartenleser und passenden Softwareanwendungen.

- Der **AAA**[17] **Proxy** repräsentiert eine logische Proxyfunktionalität, die sich in irgendeinem Netz zwischen WLAN und 3GPP AAA Server befinden kann. Diese AAA Proxies haben die Fähigkeit, die AAA Information zwischen WLAN und 3GPP AAA Server weiterzugeben. Die Anzahl der AAA-Intermediateproxies ist von den 3GPP Spezifikationen her nicht begrenzt. Diese AAA Proxyfunktionalität kann sich in einem separaten physikalischen Netzknoten befinden, sie kann im 3GPP AAA Server sein oder in irgendeinem anderen physikalischen Netzknoten. Wir halten also fest: Der AAA Proxy in einem 3GPP Netz wird 3GPP AAA Proxy genannt.
- Der **3GPP AAA** Server ist innerhalb des 3GPP Netzes angesiedelt. Er führt folgendes aus:
    - Er empfängt die Authentifikationsinformation und das Teilnehmerprofil (einschließlich der Autorisierungsinformation des Teilnehmers) vom Heimatregister/Heimatteilnehmerserver HLR/HSS des 3GPP Heimatnetzes des 3GPP Teilnehmers ab.
    - Er authentifiziert den 3GPP Teilnehmer auf der Basis der Authentifikationsinformation, die er erhalten hat. Die Authentifikationsmeldung kann dabei AAA Proxies durchlaufen.
    - Er teilt Autorisierungsinformationen an das WLAN eventuell via AAA Proxies mit.
    - Er registriert seine (d.h. des 3GPP AAA Servers) Adresse oder seinen Namen mit dem HLR/HSS für jeden authentifizierten und autorisierten 3GPP Teilnehmer.
    - Er kann auch als AAA Proxy agieren (siehe oben).
- Das lokale Heimatregister (Home Local Register) **HLR** und der Heimatteilnehmer Server (Home Subscriber Server) **HSS,** die im Heimatnetz des 3GPP Teilnehmers vorhanden sind, sind die Instanz, welche die Authentifikations- und Teilnehmerdaten enthält, die für den 3GPP Teilnehmer erforderlich sind, wenn er auf den WLAN Interworking Dienst zugreifen will.
- Das „WLAN Access Gateway“ (**WAG**) ist eine Schnittstelle, über welche Daten an/von dem WLAN Zugangsnetz **(WLAN Access Network)** weitergeleitet werden sollen via einem Public Land Mobile Network (PLMN), um ein WLAN UE mit auf Paketvermittlung (packet switched - kurz PS) basierenden 3G Diensten zu unterstützen
- Das "Online Charging System" (**OCS)** wird in 3GPP TR 32.815 [32.815] beschrieben. Es wird hier nicht weiter darauf eingegangen.
- Das **PDG** ist das „Packet Data Gateway“

Im Folgenden sind zum Verständnis der obigen Abbildung noch die Schnittstellen aufgeführt gemäß [3GPP23.234]:

- **Wa** verbindet das WLAN Zugangsnetz, eventuell über Zwischennetze, mit dem 3GPP Netz, d.h. mit dem 3GPP AAA Proxy im Roamingfall und dem 3GPP AAA

[17] AAA bedeutet Authentifizierung, Autorisierung und Abrechnung

Server im Nichtroamingfall. Das Hauptziel der Protokolle, die diese Stelle durchlaufen, ist der sichere Transport der AAA Informationen. Dieser Referenzpunkt muss auch alten WLAN Zugangsnetzen gerecht werden.

- **Wx** liegt zwischen dem 3GPP AAA Server und HSS. Die Hauptbestimmung der Protokolle, die diesen Punkt durchlaufen, ist die Kommunikation zwischen der WLAN AAA Infrastruktur und dem HSS. Die Funktionalität dieses Punktes soll folgendes ermöglichen :
    - Abfrage des Authentifikationsvektors vom HSS, z.B. für die USIM Authentifikation.
    - Abfrage der WLAN zugangsbezogenen Teilnehmerinformation (Profile) vom HSS.
    - Registrierung des 3GPP AAA Servers eines autorisierten (für den WLAN Zugang) WLAN Nutzers im HSS.
    - Anzeige von Änderungen des Teilnehmerprofils innerhalb des HSS (z.B. Anzeige der Absicht der Dienstbeendigung).
    - Ausführen der Prozeduren zwischen dem 3GPP AAA Server und dem HSS.
    - Abfrage der "online/offline" Abrechnungsfunktionsadressen vom HSS.
    - Wiederherstellungsprozedur bei der Fehlerbeseitigung zwischen dem HSS und dem 3GPP AAA Server.
    - Abfrage der Dienst bezogenen Information (z.B. W-APNs, die vom WLAN UE ausgewählt sein können), die eine Angabe einschließen, ob dem Visited Public Land Mobile Network (VPLMN) erlaubt ist, diesen Dienst bereitzustellen.
- **D'/Gr'** ist optional und liegt zwischen dem 3GPP AAA Server und dem pre-R6 HLR/HSS. Das Hauptziel der Protokolle, die diesen Punkt durchlaufen, ist die Kommunikation zwischen der WLAN AAA Infrastruktur und dem HLR. Die Protokolle, die über diesen Punkt laufen, basieren auf den D/Gr Referenzen, wie sie in 3GPP TS 29.002 [29.002] definiert sind. Die Unterstützung von D'/Gr' Referenzpunkten erfordert keine Modifikationen beim Mobile Application Part (MAP) Protokoll im HLR. Wenn es das HLR ermöglicht, ist die Funktionalität des Punktes folgende:
    - Abfrage des Authentifikationsvektors vom HLR, z.B. für die USIM Authentifikation.
    - Registrierung des 3GPP AAA Servers eines autorisierten (für den WLAN Zugang) WLAN Nutzers im HLR.
    - Anzeige von Änderungen des Teilnehmerprofils innerhalb des HLR (z.B. Anzeige der Absicht der Dienstbeendigung).
    - Ausführen der Prozeduren zwischen dem 3GPP AAA Server und dem HLR.
    - Abfrage der "online/offline" Abrechnungsfunktionsadressen vom HLR
    - Wiederherstellungsprozedur bei der Fehlerbeseitigung zwischen dem HLR und dem 3GPP AAA.
    - Abfrage der dienstbezogenen Information (z.B. W-APNs, die vom WLAN UE aufgerufen werden), die eine Angabe darüber beinhalten, ob der

Dienst vom Home Public Land Mobile Network (HPLMN) oder von einem identifizierten VPLMN zu unterstützen ist.

- Der **Wo** Referenzpunkt wird von einem 3GPP AAA Server benutzt, um mit dem 3GPP Online Charging System (OCS) zu kommunizieren. Der Hauptzweck der Protokolle, die diesen Punkt durchlaufen, ist der Transport von "online"-abrechnungsbezogenen Informationen sowie die Guthabenkontrolle für die "online"-belasteten Teilnehmer durchzuführen. Die Funktionalität von Wo ist der Transport von "online"-Abrechnungsdaten.
- Der **Wf** Referenzpunkt liegt zwischen dem 3GPP AAA Server und dem 3GPP "Charging Gateway" (CGw) mit der 3GPP "Charging Gateway Funktion" (CGF)/"Charging Collection Function" (CCF). Das Hauptziel der Protokolle, die über diesen Punkt abgewickelt werden, ist es, Abrechnungsinformationen zu den Abrechnungs-Gateways bzw. Abrechnungssammelstellen der 3GPP Operatoren, die im besuchten Netz oder im Heimnetz des Teilnehmers liegen, weiterzuleiten.
- **Wg** ist eine AAA Schnittstelle zwischen dem 3GPP AAA Server/Proxy und dem WAG. Sie wird verwendet, um Informationen, welche vom WAG benötigt werden, um die Strategie für autorisierte Teilnehmer durchzusetzten.
- **Wn** liegt zwischen dem WLAN Zugangsnetz (WLAN Access Network) und dem WAG. Dieses Interface soll den Verkehr auf einem durch das WLAN Nutzergerät UE initiierten Tunnel über das WAG leiten. Es gibt mehrere unterschiedliche Wege, dieses Interface zu implementieren. Welche spezifischen Methoden für dieses Interface implementiert werden, bestimmt eine lokale Vereinbarung zwischen WLAN Zugangsnetz (WLAN AN) und der PLMN.
- **Wp** liegt zwischen WAG und dem Packet Data Gateway (PDG).
- **Wi** liegt zwischen dem Packet Data Gateway (PDG) und einem Datenpaketnetz. Das Datenpaketnetz kann ein externer öffentlicher Operator oder ein privates Datenpaketnetz oder ein interner Datenpaketnetz-Operator sein, wie z. B. Eingangspunkt eines IP Multimedia Subsystem (IMS), ein Netz mit RADIUS Abrechnung oder Authentifizierung oder ein Netz mit welchem das Dynamic Host Configuration Protokoll (DHCP) abgewickelt wird. Der Wi Punkt ist dem Gi Punkt, der von einer PS Domäne zur Verfügung gestellt wird, ähnlich. Die Zusammenarbeit mit Datenpaketnetzen ist über den Wi Punkt vorgesehen, basierend auf IP. Von mobilen Terminals angebotene Dienste via Wi dürfen global durch das öffentliche Adressenschema des Operators oder durch die Benutzung eines privaten Adressenschemas adressierbar sein.
- **Wm** liegt zwischen dem 3GPP AAA Server und dem PDG. Die Funktionalität dieses Punktes erlaubt folgendes:
    - Der 3GPP AAA Server kann Tunnelungsattribute und IP Konfigurations-parameter von WLAN UEs über das Datenpaket Gateway abfragen.
    - Es können Nachrichten für die Diensteauthentifizierung dem WLAN UE and dem 3GPP AAA Server weitergeleitet werden.
    - Es können Nachrichten für die Diensteautorisierung zwischen PDG und 3GPP AAA Server weitergeleitet werden.
    - Es können Authentifikationsdaten zum Zwecke der Tunneletablierung, Tunneldatenauthentifizierung und Verschlüsselung weitergeleitet werden.

- Der Punkt **Wd** verbindet den 3GPP AAA Proxy, möglicherweise über Zwischennetze mit dem 3GPP AAA Server. Das Hauptziel der Protokolle, die über diesen Punkt abgewickelt werden, ist der sichere Transport von AAA Informationen. Die Funktionalität von Wd schließt folgendes ein:
  - Datentransport zur Authentifizierung zwischen dem 3GPP AAA Proxy und dem 3GPP AAA Server.
  - Datentransport zur Autorisierung zwischen dem 3GPP AAA Proxy und dem 3GPP AAA Server.
  - Abrechnungsdaten für jeden WLAN Nutzer übertragen.
  - Übertragten von Schlüsseldaten zum Schutz der Integrität und zur Verschlüsselung der Luftschnittstelle.
  - Weiterleitung von Authentifizierungsdaten zum Zwecke des Tunnelaufbaus, Tunneldatenauthentifizierung und Verschlüsselung.
  - Einen Nutzer vom WLAN Zugang trennen zur sofortigen Beendigung eines Dienstes.
  - Die Fähigkeit, die Operatornetze zu identifizieren, zwischen denen das Roaming auftritt.
- **Wu** liegt zwischen dem WLAN Nutzergerät UE und dem PDG. Wu repräsentiert den vom WLAN UE-initiierten Tunnel zwischen dem WLAN UE und dem PDG. Der Verkehr für das Wu Referenzpunkt Protokoll wird durch die Punkte Wn und Wp bereitgestellt, welche sicherstellen, dass die Daten über die **WLAN** Verbindungsschnittstelle **WAG** geleitet werden, wo das Routing durchgeführt wird. Die Funktionalität des Wu Punktes liefert folgende Eigenschaften:
  - WLAN UE-initialisierter Tunnelaufbau
  - Übertragung von Datenpaketen des Nutzers innerhalb des vom WLAN UE initiierten Tunnels.
  - Abbauen des WLAN UE initiierten Tunnels
- Der **Ww** Referenzpunkt verbindet das WLAN Nutzergerät UE mit dem WLAN Zugangsnetz gemäß IEEE 802.1x Spezifikation.

Man muss feststellen, dass die Interworking Architektur prinzipiell zwar Roaming – d.h. den Wechsel zwischen WLAN und 3G Netzen - ermöglicht, allerdings wird beim Wechsel zwischen den Netzen dabei jedes Mal eine neue IP-Adresse vergeben. Ein übergangsloses („seamless“) Roaming ist also nicht möglich. Hierzu ist der Einsatz weiterer Mechanismen notwendig. Ein Mechanismus, der die Möglichkeit die IP-Adresse beizubehalten in dem Sinne bietet, dass ein Mobiles Endgerät über dieselbe IP Adresse – seine Heimatadresse - erreichbar ist, ist Mobile IP wie in Abschnitt 2.3.1 dargestellt.

2.4.4 Roaming durch Mobile IP

Um während eines Handovers bzw. Handoffs die bestehenden Sitzungen und ihre Verbindungen zu erhalten, einen Handover also übergangslos durchzuführen, muss bei loser Kopplung das Routing an die neue IP-Adresse angepasst werden.
MIP stellt für die gesamte Dauer der Verbindung eines einen Mobile Node darstellenden Endgerätes eine IP Adresse zu Verfügung. Die genaue Funktionsweise von MIP wurde bereits in Abschnitt 2.3.1 dargestellt. Beim Roaming zwischen UMTS und WLAN wird

der HA im Netz des Mobilfunkbetreibers angesiedelt, mit dem der Endnutzer einen Vertrag hat. Dieses Netz ist das Heimnetz. Bewegt sich der Endnutzer in ein Fremdnetz, d.h. in ein WLAN oder zu einem anderen Mobilfunkbetreiber, so wird sein Heimatagent (HA) über die neu zugewiesene Adresse, die Care-of Address (COA) vom Fremdagenten (FA) des Fremdnetzes informiert. Möchte nun ein anderer Nutzer (CN, Correspondent Node) den mobilen Nutzer MN kontaktieren, so schickt er Daten an die feste Adresse des MN. Der Heimagent tunnelt diese Pakete an die aktuelle COA des MN zum Fremdagenten und stellt dem MN das ausgepackte Paket zu. Dieser antwortet an die Adresse des CN.
Ein Problem dieses Ansatzes ist das bereits in Abschnitt 2.3.1 beschriebene trianguläre Routing, das vor allem bei großen Entfernungen zwischen Heimnetz und Fremdnetz (bzw. bei geringen Entfernungen zwischen MN und CN) zu überflüssigem Verkehr im Internet führt. Außerdem entsteht ein Overhead durch das Tunneln der Pakete. Durch Optimierungen lässt sich der CN über die neue Position des MN informieren, so dass das Routing direkt erfolgt. Eine weitere Optimierung sieht den Kontakt zwischen altem und neuem FA bei einem Handover vor, dies ist für eine weiche Sitzungsübergabe erforderlich, da der Handover-Delay sonst zu groß werden würde und auch eine darauf aufbauende TCP-Verbindung trotz Toleranz abbrechen würde, die Sitzung also beenden würde. Dennoch hat MIP verhältnismäßig große Handover Latenzen, ist also nur für nicht Echtzeit-Applikationen geeignet.
Daher gibt es noch weiter Protokolle für Mikromobilität, wie Cellular IP, Hierarchical MIPv6 oder Hawaii, welche Mobile IP ergänzen. Sie sollen die Last der häufigen Aktualisierungen von Heim- und Fremdagenten durch neue Registrierungen und Bindungen dämpfen, indem sie lokale Wechsel innerhalb einer Domäne verbergen. Dadurch wird auch der vertikale Handover optimiert [Schiller03]. Da es in dieser Arbeit primär um die Informationssicherheit geht, wird auf eine detaillierte Darstellung dieser Protokolle hier verzichtet.

2.5 Public Key Infrastruktur (PKI)

Da der Einsatz von Public Key Infrastrukturen für die in dieser Arbeit entwickelte Architektur wesentlich ist, werden an dieser Stelle zunächst die Grundlagen einer PKI erläutert:

Eine Public Key Infrastructure (PKI) enthält für gewöhnlich als grundlegende Komponenten eine Registration Authority (RA), eine Certification Authority (CA) und ein Repository, welches öffentlich Zugang zu Zertifikaten und Zertifikatssperrlisten (CRLs[18]) – für Nutzer. Die allgemeine Architektur einer PKI ist in der folgenden Abbildung 23 dargestellt.

[18] certificate revocation lists oder Sperrlisten

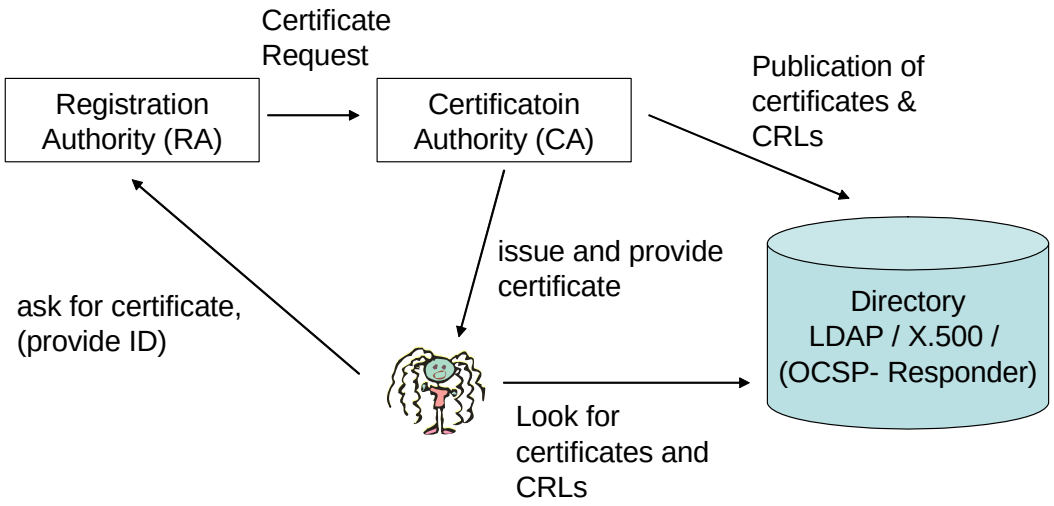


Abbildung 23: Grundlegende PKI Komponenten

Zuerst muss sich eine Entität, wie z. B. der Nutzer bei einer Registrierungsinstanz - der RA also - registrieren. Der Nutzer fragt nach einem Zertifikat und gibt der RA die notwendige Identifizierungsinformation, wie z. B. seinen Namen und weitere relevante persönliche Informationen. Die RA überprüft die Informationen, die vom Nutzer erhalten wurden und veranlasst die CA, ein Zertifikat für den erfolgreich identifizierten Nutzer auszustellen. Die CA stellt die Zertifikate aus und verteilt sie an die Nutzer. Die CA kann die ausgestellten Zertifikate auch veröffentlichen. Um mit zurückgerufenen Zertifikaten zurecht zu kommen, veröffentlicht die CA Informationen darüber, welche Zertifikate zurückgerufen wurden, in so genannten Zertifikatssperrlisten (CRL), die anderen zugänglich sein sollten, damit sie den Status der Zertifikate überprüfen können. CRL Informationen können z. B. mit einem Lightweight Directory Access Protocol (LDAP) Server öffentlich zu Verfügung gestellt werden. In den Zertifikaten wird die Zugehörigkeit von öffentlichen Schlüsseln zu bestimmten Identitäten von Nutzern bestätigt. Die Zertifikate sind von der sie ausstellenden CA signiert.
Nutzer vertrauen dabei Zertifikaten, die von einer Instanz signiert sind, deren Signatur sie verifizieren können und der sie vertrauen. Typischerweise ist dies z. B. die CA, die auch ihr eigenes Zertifikat ausgestellt hat. Wenn die Verifikation einer gegebenen Signatur ein positives Ergebnis liefert und wenn das zugehörige Zertifikat nicht in einer Revokationsliste steht, dann gelten die Signatur und das zugehörige Zertifikat als vertrauenswürdig. In [Linn00] werden verschiedene Vertrauensmodelle für PKIs gegenübergestellt. Typischerweise gibt es mehrere CAs, welche in einer hierarchischen Struktur angeordnet sind. In Kapitel 6 wird dann die PKI basierte Lösung, welche im Rahmen dieser Arbeit entwickelt wird, detailliert beschrieben.
Eine prinzipielle Alternative zu einem hierarchischen PKI System ist das „Web of Trust“. Im Folgenden Abschnitt wird dieses Prinzip beschrieben und es wird erläutert, warum es für die im Rahmen dieser Arbeit entwickelte Lösung nicht eingesetzt wird.

2.5.1 Netz des Vertrauens

Die Schlüsselverwaltung in einem Netz des Vertrauens („Web of Trust“) erfolgt mit Hilfe von Schlüsselringen. Im „Public Keyring“ eines Nutzers werden alle öffentlichen Schlüssel und die zugehörigen Zertifikate gespeichert. Im „Private Keyring“ befinden sich die dem Nutzer eigenen privaten Schlüssel. Jeder Nutzer ordnet den ihm bekannten öffentlichen Schlüsseln einen Vertrauenswert zu, der das Vertrauen in dessen Besitzer ausdrückt („Owner Trust“). Den „Owner Trust“ Wert legt ein Nutzer einzeln für alle in seinem „Public Keyring“ befindlichen Schlüssel fest. Er hat die Möglichkeit fünf verschiedene Werte zuzuordnen:

1. Der Wert „unknown“ gilt für Nutzer, über die man keine weiteren Informationen hat.
2. Der Wert „not trusted“ gilt für Nutzer, denen nicht vertraut wird.
3. Der Wert „marginal“ ist für Nutzer, denen nicht voll vertraut wird
4. Der Wert „complete“ steht für Nutzer, denen voll vertraut wird, und
5. der Wert „ultimate“ ist für Nutzer gedacht, deren Private Key sich im eigenen „Private Keyring“ befindet.

Daraus wird der Grad des Vertrauens in die Authentizität anderer Schlüssel, der „Key Legitimacy“ Wert abgeleitet.

Wenn Nutzer A den öffentlichen Schlüssel von Nutzer B signiert und diese Signatur anschließend an einen Keyserver überträgt, dann kann diese Signatur von einem Nutzer C zur Beurteilung der Authentizität des öffentlichen Schlüssels von Nutzer B eingesetzt werden. Nutzer C prüft dabei, ob er den öffentlichen Schlüssel von Nutzer A selbst signiert hat, und ob er ihm den „Owner Trust“-Wert „marginal“ oder „complete“ zugeordnet hat. Ist das der Fall, so erhält die Signatur von Nutzer A genau diesen Wert als „Signatory Trust“ Wert. Hat Nutzer C den Schlüssel von Nutzer B selbst signiert, so erhält diese Signatur den Signatory Trust Wert „complete“; in allen anderen Fällen wird der Signatur der Wert „not trusted“ zugeordnet. Der „Signatory Trust“-Wert wird dabei einer Signatur und nicht einer Person zugeordnet.

Das Vertrauen in die Authentizität eines öffentlichen Schlüssels wird durch den „Key-Legitimacy“-Wert L ausgedrückt. Er wird aus dem Signatory Trust der signierenden Schlüssel wie folgt berechnet:

$$L = \frac{m}{M} + \frac{c}{C}, \text{ wobei}$$

m die Anzahl von Signaturen ist, deren „Signatory Trust“ Wert „marginal“ ist, und c die Anzahl von Signaturen ist, deren Signatory Trust „complete“ ist. M ist die Anzahl von Signaturen mit einem „Signatory Trust“ Wert „marginal“, die erforderlich ist, damit ein Schlüssel als authentisch eingestuft wird, und C ist die Anzahl von Signaturen mit einem „Signatory Trust“ Wert „complete“, die erforderlich ist, damit ein Schlüssel als authentisch eingestuft wird.

Wenn L = 0 gilt, dann wird der überprüfte Schlüssel als nicht authentisch angesehen. Wenn L zwischen 0 und 1 liegt, wird er als „teilweise authentisch“ angesehen und wenn L > 1 oder L=1 gilt, dann als „vollkommen authentisch“. Die Werte für M und C kann dabei jeder Nutzer frei wählen.

Zertifizierung nach dem Modell eines „Web of Trust“ bietet nach [Fed06] zwar den Vorteil einer einfachen flexiblen Nutzung, allerdings ist der Nachteil eine schwer erreichbare Beweisführung im Streitfall. Außerdem ist das finden eines vertrauenswürdigen Pfades aufwendiger als bei hierarchischer Zertifizierung [Fed06]. Die Revokation von Zertifikaten ist auch nicht sofort allgemein bekannt, wie es i. d. R. bei einer PKI der Fall ist. Auf Grund dieser Nachteile wird für die in dieser Arbeit entwickelte Roaming Lösung kein „Web of Trust“ eingesetzt.

2.6 RoleBased Access Control (RBAC)

Da in Kapitel 7 bei der in dieser Arbeit entwickelten Autorisierungslösung eine rollenbasierte Zugriffskontrolle zum tragen kommt, wird im nächsten Abschnitt das RBAC Modell nach [FeKuCh03] beschrieben.

RBAC [RBAC] ist ein Verfahren zur Zugriffskontrolle für Dateien oder Dienste in Rechnernetzen und Mehrbenutzersystemen. Es ist das wichtigste Modell für Zugriffskontrolle geworden, da es die Komplexität und die Kosten für die Administration von Zugriffsrechten in großen Netzen reduziert. Bei RBAC werden Nutzern Rollen zugeordnet. Da mehrere Nutzer die gleiche Rolle haben können, werden so automatisch Gruppen gebildet. Typische Beispiele für Rollen sind Systemadministrator, Webmaster, Netzadministrator oder einfach Nutzer. Die Zuordnung von Nutzern zu Rollen basiert immer auf der eindeutigen Identifikation des Nutzers und kann daher keine Anonymität unterstützen. Sie ermöglicht es nicht nur individuell für einen Nutzer Rechte zu vergeben, sondern auch für ganze Gruppen Zugriff auf bestimmte Ressourcen zu geben oder zu sperren.

Die formale Beschreibung von RBAC nach [FeGiLy92] lautet wie folgt:

- Für jedes Subjekt ist die aktive Rolle diejenige Rolle, welche es gerade verwendet: *AR(s:Subjekt)={die aktive Rolle für Subjekt s}*
- Jedes Subjekt kann autorisiert sein, eine oder mehrere Rollen auszufüllen: *RA(s:Subjekt)={Autorisierte Rollen für Subjekt s}*
- Jede Rolle kann autorisiert sein, eine oder mehrer Aktionen durchzuführen: *TA(r:Rolle)={Aktionen, für die die Rolle r autorisiert ist.}*
- Subjekte können Aktionen ausführen. Das Prädikat exec(s,a) ist genau dann wahr, wenn das Subjekt s zum gegenwärtigen Zeitpunkt die Aktion a ausführen kann. Andernfalls ist das Prädikat falsch:

  exec(s:Subjekt,a:Aktion)={wahr, falls das Subjekt s die Aktion a durchführen kann}

  1. Rollenzuordnung: Ein Subjekt kann eine Aktion nur ausführen, wenn dem Subjekt eine Rolle zugeordnet ist:
     $\forall s : Subjekt, a : Aktion \cdot exec(s,a) \Rightarrow AR(s) \neq \phi$
  2. Rollenautorisierung: Ein Subjekt muss autorisiert sein für seine aktive Rolle: $\forall s : Subjekt \cdot AR(s) \subseteq RA(s)$

3. Aktionsautorisierung: Ein Subjekt kann eine Aktion nur ausführen, wenn die aktive Rolle des Subjekts dafür autorisiert ist:
   $\forall s : Subjekt, a.Aktion \cdot exec(s,a) \Rightarrow a \in TA(AR(s))$

Die zentralen Terme bei RBAC sind Nutzer, Rolle und Zugriffsrecht. Daher gibt es bei RBAC eine Menge von Subjekten, wie z. B. Nutzern, S, eine Menge von Rollen R und eine Menge von Zugriffsrechten (permissions) P.

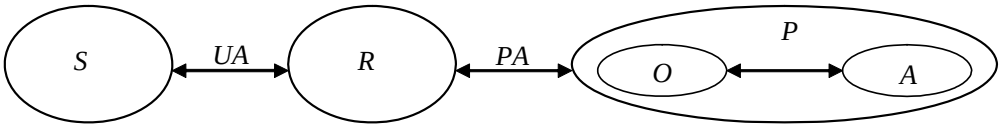


Abbildung 24: RBAC Grundmodell

Für Elemente dieser Mengen gibt es zwei Verknüpfungsrelationen $UA \subseteq S \times R$ und $PA \subseteq P \times R$ wie in Abbildung 24 gezeigt. UA definiert eine Relation zwischen Nutzer und Rolle, wohingegen PA eine Relation zwischen Zugriffsrechten und Rollen definiert. Beides sind n zu n Relationen. Die Typen von Aktivitäten hängen vom Typ des Systems ab, bei dem die Autorisierung betrachtet werden muss.

Wenn ein Nutzer $s \in S$ von seiner alten in eine neue Kategorie eingeordnet werden soll, wird ihm lediglich eine neue Rolle $r_{new} \in R$ zugeordnet. Dadurch verliert er alle Zugriffsrechte, die er in seiner alten Rolle $r_{old} \in R$ inne hatte und erhält alle einer neuen Rolle entsprechenden Zugriffsrechte.

Ein weiterer Vorteil des RBAC Modells ist die Möglichkeit hierarchische Strukturen mittels Vererbung aufzubauen. Rollenhierarchien sind sinnvoll, da i. d. R die Rechte, die unterschiedlichen Rollen z. B. in Unternehmen zugeordnet sind, sich gegenseitig überlappen. Die Möglichkeit der Vererbung von Rechten verringert den Administrationsaufwand im Vergleich zu der Möglichkeit für jede spezifische Rolle alle Rechte zu definieren. I. d. R. gibt es viele allgemeine Rechte, wie z. B. der Zugriff auf E-Mail oder irgendwelche allgemeinen Server, welche vielen Rollen zuzuordnen sind. Die Definition einer allgemeinen Rolle, welche über diese Rechte verfügt von der dann andere Rollen die Rechte erben können ist effizienter als die Zuordnung dieser Rechte zu allen spezifischen Rollen getrennt.

Formal sind Rollenhierarchien bei RBAC generell wie folgt definiert:
Eine Rollenhierarchie $RH \subseteq R \times R$ ist eine Halbordnung auf der Menge der Rollen, welche als Vererbungsbeziehung $\succeq$ geschrieben wird, wobei $r_1 \succeq r_2$ gilt, wenn

$authorized\_permissions(r_2) \subseteq authorized\_permissions(r_1)$ und

$authorized\_users(r_1) \subseteq authorized\_users(r_2)$

Dabei gilt:[19]
authorized_permissions(r:ROLLEN): $\rightarrow 2^P$ ist die Zuordnung einer Rolle r zu einer Menge von Zugriffsrechten: $authorized\_permissions(r) = \{p \in P \mid r' \preceq r, (p, r') \in PA\}$
authorized_users(r:ROLLEN): $\rightarrow 2^S$ ist die Zuordnung einer Rolle r zu einer Menge von Nutzern: $authorized\_users(r) = \{s \in S \mid r' \succeq r, (s, r') \in UA\}$

RBAC wird typischerweise dafür eingesetzt, um Zugriffshierarchien in Unternehmen abzubilden. In Kapitel 7 sind Beispiele gegeben, die dies verdeutlichen. Dort ist beschrieben, wie das oben dargestellte RBAC Modell im Rahmen der in dieser Arbeit entwickelten Architektur angewandt wird.

2.7 Andere Forschungsprojekte

In diesem Kapitel werden andere Roaming Modelle aus Forschungsprojekten beschrieben, die ähnliche Funktionalitäten wie das in dieser Arbeit entwickelte Modell, aufweisen und es werden die Unterschiede zu dem in dieser Arbeit entwickelten Modell aufgezeigt.

2.7.1 4GPlus

4GPLUS: (4th Generation Platform Launching Ubiquitous Services)
Partner: Telematica Instituut, Lucent Technologies, KPN
Ziele: Dienstekontroll-Lösung für "seamless roaming" zwischen heterogenen Netzen (WLAN, 3G mobile, enterprise,..)

Die folgende Abbildung 25 zeigt die Architektur von 4GPlus.

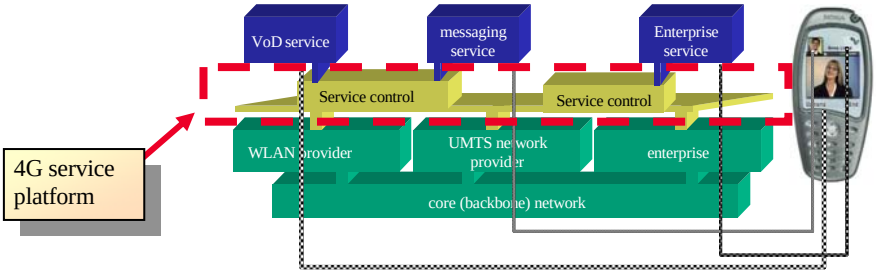


Abbildung 25: 4GPlus Architektur, Quelle: [http://www.freeband.nl/kennisimpuls/projecten/4gplus/ENindex.html]

Im Rahmen von 4GPlus gibt es seinen „Seamless Roaming Demonstrator":

- Er ermöglicht das Messen des Netz (WLAN) Signals

[19] Für eine Menge X, wird die Mächtigkeit von X mit 2^X angegeben.

- Umherwandern von Endgeräten (WLAN) zwischen unterschiedlichen Zugangsnetzen in verschiedenen Verwaltungs-Domänen.
- Minimale (grundlegende) Unterstützung für Roaming Profile/ preference settings

4GPlus ist charakterisiert durch folgende Eigenschaften:

- Eingesetzte Technologien: WLAN Authentifikation (802.1x, Zertifikate, RADIUS), Mobile IP, Session Initiation Protocol (SIP)
- Grenzen: Nur WLAN und GPRS Unterstützung, kein Single Sign-On; Keine PKI, keine Autorisierung

Defizite von 4GPlus im Vergleich zur Lösung aus dieser Arbeit:

- Beidseitige Authentifikation
  - 4GPlus benutzt beidseitige Zertifikats-basierte Authentifikation
  - Problematik der Unterstützung von offline Nutzern wurde nicht beachtet bei 4GPlus
- Nicht-Abstreitbarkeit für komplexe Geschäftsmodelle: Nicht-Abstreitbarkeit wird als Eigenschaft erwähnt, Es sind jedoch keine Details verfügbar. Geschäftsmodelle werden nicht berücksichtigt.
- PKI Server basierte Zertifikatspfad Überprüfung im Sinne des RSP Modells:
  - 4GPlus leitet komplexe Broker-artige Modelle ab (service aggregators, network integrators). Mit dem Problem der dem Überprüfer unbekannten Aussteller CAs befasst man sich in 4Gplus nicht.
  - Auf die Überprüfung von komplexen Zertifikatepfaden wird im Rahmen von 4G+ nicht eingegangen
- Integration einer Unternehmens-PKI: Dieser Aspekt wird bei 4GPlus nicht betrachtet.
- Zertifikatebasierte Autorisierung:
  - Information darüber wie 4GPlus das Problem des Austauschs von Autorisierungsinformation löst, ist nicht gegeben
  - Zugriffskontrolle ist als Sicherheitsziel erwähnt, es sind aber keine Details dargestellt.
- Hochgeschwindigkeitsauthentifizierung: wird nicht bedacht oder erwähnt

2.7.2 SAG in der Evolute Architektur

EVOLUTE (seamless multimedia services over all IP-based infrastructures)
Partner: Intracom, Motorola, ALCATEL, Fraunhofer-FOKUS, Cerfriel, UNIS, TELLA
Ziele: All-IP basierte Netz Infrastruktur für „seamless roaming“ basierend auf WLAN und UMTS Technologien

Die Authentifikation ist in EVOLUTE SIM basiert. Der Nutzer, der in das WLAN eintritt ist einerseits authentifiziert bezüglich der WLAN Signalisierungsprozeduren andererseits gleichzeitig auf Basis seiner SIM Karte. Die folgende Abbildung 26 zeigt das SIM Access Gateway (SAG) im Zusammenspiel mit den anderen Komponenten.

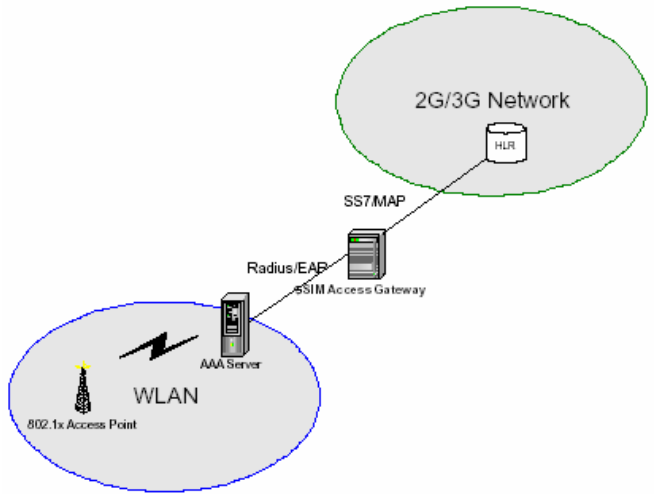


Abbildung 26: SAG

Die Authentifikation funktioniert wie folgt:

1. Ein mobiler Nutzer betritt eine WLAN Domäne. Der MT informiert den Access Point/Router (802.11 Technologie), dass er Zugang zum örtlichen Netz zu erhalten wünscht.
2. Der Access Point/Router kontaktiert den AAA Server
3. Der AAA Server kontaktiert das SAG um den Nutzer zu authentifizieren auf Basis seiner SIM.
4. Der SAG tauscht die Authentifikationsinformation mit dem HLR aus.
5. Der SAG abhängig vom Ergebnis der vorherigen Prozedur schickt dem AAA eine Erfolgs/Fehler – Meldung zurück.
6. Der AAA sendet das Ergebnis zurück zu dem mobilen Endgerät über den Access Point/Router

Die folgende Abbildung 27 zeigt den Nachrichtenfluss der SIM basierten Authentifikation beim WLAN.

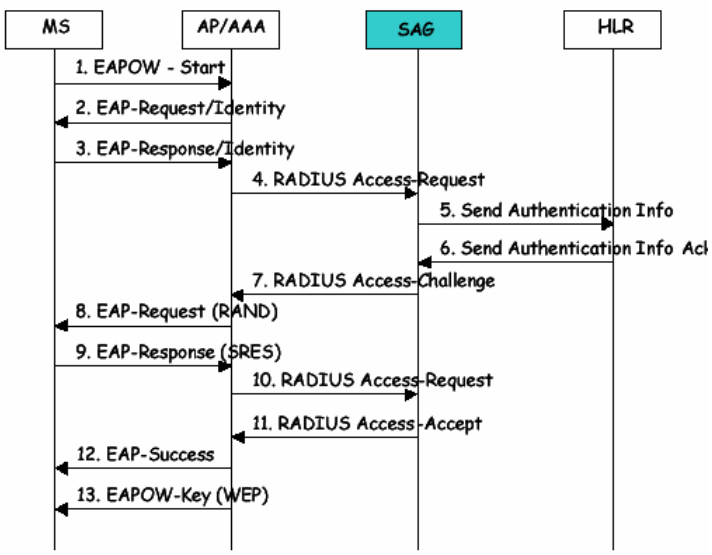


Abbildung 27: SIM basierte Authentifikation für WLAN

Vergleich der Lösung aus dieser Arbeit mit Evolute:

- Beidseitige Authentifikation: Keine beidseitige Authentifikation in „Evolute“
- Nicht-Abstreitbarkeit für komplexe Geschäftsmodelle: Keine Nicht-Abstreitbarkeit in Evolute, da keine Anwendung von „Public-Key“ Kryptographie
- PKI Server basierte Zertifikatspfadvalidierung: In Evolute nicht notwendig, da keine Zertifikate eingesetzt werden.
- Integration einer Unternehmens-PKI: Nicht sinnvoll, da keine Zertifikate vorgesehen sind.
- Zertifikatebasierte Autorisierung:
    - Autorisierung ist nicht zertifikatebasiert, da keine Zertifikate in diesem Projekt verwendet werden.
    - Autorisierungsinformation wird in einer für Authentifizierung und Autorisierung gemeinsamen Antwort erhalten, nachdem eine Anfrage zum HLR gesendet wurde zwecks Authentifikation.
- Hochgeschwindigkeitsauthentifikation wird in Evolute unterstützt. SIM-basierte Authentifikation kann schnell sein, auch wenn dies nicht explizit erwähnt wird.

2.7.3 Moby Dick

Das Mobility and Differentiated Services in a Future IP Network (Moby Dick) hat gemäß [http://www.ist-mobydick.org/] folgende Ziele:[20]

- To facilitate the development of seamless access to existing and emerging IP-based applications.

[20] Um Fehler durch eine Übersetzung zu vermeiden, wird der englische Originalwortlaut übernommen.

- To propose an architecture for wireless Internet access by developing new mechanisms for seamless hand-over, QoS support after and during hand-over, AAA, and charging.
- To facilitate new business opportunities for operators, manufacturers, services providers, and content providers for wireless, access, and backbone technology and services.
- To contribute actively to standardisation bodies, such as Internet Engineering Task Force and Internet Research Task Force.

Dabei wird gemäß [http://www.ist-mobydick.org/] folgender technischer Ansatz verfolgt:
“In order to continue to evolve 3rd Generation mobile and wireless infrastructure towards the Internet - targeting IST 2000 IV 5.2 "Terrestrial Wireless System and Networks", the project Moby Dick will define, implement, and evaluate an IPv6-based mobility-enabled end-to-end QoS architecture starting from the current IETF's QoS models, Mobile-IPv6, and AAA framework. A representative set of interactive and distributed multimedia applications will serve to derive system requirements for the verification, validation, and demonstration of the Moby Dick architecture in a testbed comprising UMTS, 802.11 Wireless LANs and Ethernet. In case the existing applications or the underlying architectures do not provide what is required, the necessary modification will be undertaken.”

Die wesentlichen Eigenschaften sind die folgenden [http://www.ist-mobydick.org/]:

- Definition of a common architecture integrating QoS, IPv6 mobility, and AAA (out of the separate architectural approaches for each component currently provided by the IETF) with respect to wireless issues.
- Implementation and evaluation of an IPv6-based end-to-end technological approach to fulfil the requirements of present and future mobile communication services.
- Implementation and evaluation of QoS models (e.g. Differentiated Services) in highly dynamic and heterogeneous network topologies (understanding of QoS models is normally restricted to relatively static environments).
- Definition of a suitable charging concept which would enable permanent mobile IP based services on a large scale (a strong requirement related to AAA, but currently not a topic within the IETF).
- Trans-European trial to test the implementation by using SOKRATES-ERASMUS exchange students as test-users.

Die zugrunde gelegten Netze werden bei Moby Dick stark vereinfacht. Einziges Funk-Interface ist W-CDMA. Andere Elemente, wie RNC, HLR, VLR, MSC, SGSN, GGSN, u. s. w. wurden nicht betrachtet oder durch IP-basierte Äquivalente ersetzt.
Bei Moby Dick wird “Hierarchical Mobile IPv6 Mobility Management” gemäß RFC 4140 und “Fast Handovers for Mobile IPv6” gemäß RFC 4068 angewandt.

2.7.4 DAIDALOS

"*Designing Advanced network Interfaces for the Delivery and Administration of Location independent, Optimised personal Services*" (DAIDALOS) ist ein Nachfolgeprojekt von Moby Dick. [21]
DAIDALOS [http://www.ist-daidalos.org/] "is an Integrated Project in the Thematic Priority 'Information Society Technologies' of EU Framework Programme 6 for Research and Development, which is currently in its second phase (phase 1: IST-2002-506997, phase 2: IST-2005-026943). The project budget of DAIDALOS phase 1 amounted to € 25.7 million, of which € 14.7 million was funded by the European Commission. Phase 1 had 46 partners from industry and academia, and a duration of 2.5 years. The budget of DAIDALOS phase 2 is € 22.1 million, of which € 13.8 million is funded by the European Commission. There are 38 partners from industry and academia involved in Phase 2 of the project. Phase 2 is planned to conclude by the end of 2008."

Ziel von DAIDALOS ist gemäß [http://www.ist-daidalos.org/] "to develop and demonstrate an open architecture based on a common network protocol (IPv6)"
Dies bedeutet im einzelnen [http://www.ist-daidalos.org/daten/overview/overview.htm]:

- To Design, prototype and validate the necessary infrastructure and components for efficient distribution of services over diverse network technologies beyond 3G,
- To Integrate complementary network technologies to provide pervasive and user-centred access to these services,
- To Develop an optimized signalling system for communication and management support in these networks,
- To Demonstrate the results of the work through strong focus on user-centered and scenario-based development of technology

Eine "Intelligent Interface Selection" Technik soll ohne den Anwender zu belasten zwischen einem WLAN, einem Mobilfunknetz oder Bluetooth automatisch umschalten. Eine derartige Technik ist wünschenswert und ließe sich auch bei der in die in dieser Arbeit entwickelten Architektur sinnvoll einsetzten, wenn auf den Endgeräten vorhanden.

2.7.5 Ambient Network Security Architecture

Im "Ambient Networks" Projekt " wird ein vollständiges, kohärentes drahtloses Netzwerk entwickelt, das über ein dynamisches Zusammenspiel verschiedener Netzwerke den Zugang zu jedem dieser Netzwerke durch einen sofortigen Aufbau von Vereinbarungen zwischen diesen Netzen verspricht [www.ambient-networks.org].

Es geht dabei nicht, um einen umherwandernden Nutzer, der Zugang zum Internet über verschiedene Zugangsnetze erhalten kann, wie bei der in dieser Arbeit entwickelten Architektur, sondern um die Verbindung unterschiedlicher Netze. Eine zentrale Rolle in der Architektur spielen die in diesem Projekt neu eingeführte keinem Standard entsprechenden so genannten „Negotiator"-Einheiten, welche dazu eingesetzt werden

[21] Um Fehler durch eine Übersetzung zu vermeiden, wird der englische Originalwortlaut übernommen.

sollen, die AAA Infrastruktur zu konfigurieren, so dass ein Interworking zwischen verschiedenen Netzen möglich wird, wie in folgender Abbildung dargestellt.

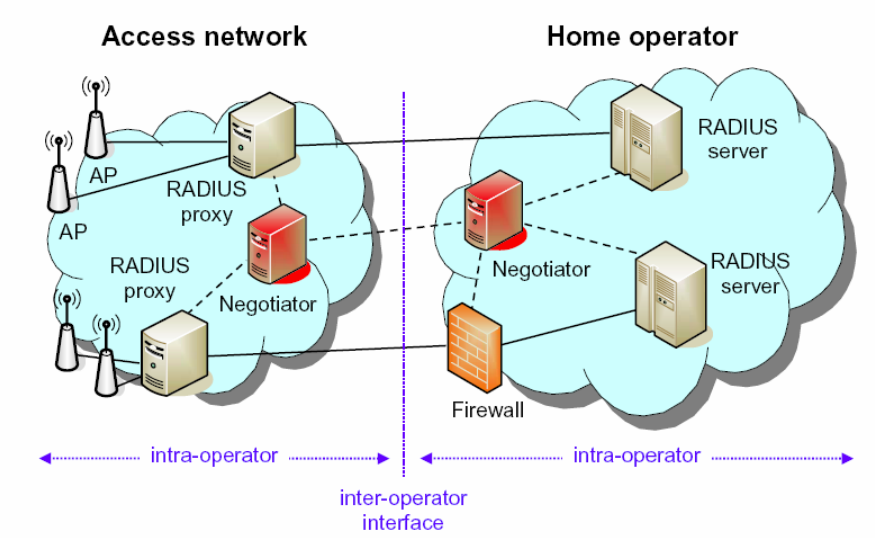


Abbildung 28 AAA Configuration Protocol [ANSA06]

Die grundlegende Authentifizierungsprozedur der „Ambient Networks Security Architecture“ (ANSA) [ANSA05] beinhaltet einen Austausch von 5 Nachrichten, welche eine Diffie-Hellmann Prozedur beinhalten. Sie ist im Wesentlichen für Home und Small Office Lösungen gedacht, wie dies aus [ANSA05] hervorgeht.[22] Die ANSA beinhaltet des Weiteren keine PKI im Gegensatz zu der in meiner Arbeit entwickelten Architektur.

Das in dieser Arbeit in Kapitel 5.1.1.1 vorgeschlagene grundlegende Authentifizierungsprotokoll kommt daher mit 3 Nachrichten aus, was einen Geschwindigkeitsvorteil gegenüber der Lösung aus dem ANSA-Projekt bedeuten dürfte.

Anders als bei meinem Ansatz werden beim „Ambient Network Security Architecture“ Projekt keine Geschäftsmodelle zugrunde gelegt, um die Interessen der involvierten Parteien zu beachten und daraus resultierende Anforderungen für die Architektur zu ermitteln. Es ist zwar ein Workshop zum Thema „Business Models“ geplant. Dieser hat jedoch bis jetzt (1. Dezember 2006) nicht stattgefunden.

Das Projekt soll sich über 3 Phasen erstrecken. Lediglich die erste Phase ist bisher abgeschlossen. In ihr wurde das Gesamtkonzept entwickelt. Die zweite Phase hat am 1.1.2006 begonnen. Sie soll am 31.12.2007 abgeschlossen sein. Dabei soll „implementation, integration, measurements and performance evaluation“ [www.ambient-networks.org] erfolgen.

[22] „Typically, this can be done in very small networks, such as home or at small offices“[ANSA05].

2.8 Ergebnis

Wegen der in Kapitel 2.2 „Drahtlose Netzwerktechnologien“ genannten Gründe wird für die in dieser Arbeit entwickelte Roaming Lösung eine zertifikatebasierte Authentifizierung vorgesehen. Die unterschiedlichen in Kapitel 2.2 beschriebenen Sicherheitseigenschaften der möglichen Zugangsnetze zum einen und die Tatsache, dass die jeweiligen Sicherheitsmechanismen nur zwischen Endgerät und Zugangspunkt greifen, während die Kommunikation über diesen Zugangspunkt hinaus prinzipiell nicht gesichert ist, erfordern die Möglichkeit, falls von einem Nutzer verlangt, zusätzliche Mechanismen zur Gewährleistung der Vertraulichkeit vorzusehen. Wie dies mittels eines hierfür geeigneten Standardprotokolls (IPSec) möglich ist, ist in Kapitel 2.3.2 „IPSec“ beschrieben.
In Kapitel 2.3 „Protokolle“ wird gezeigt, wie das übergangslose Roaming mittels Standardprotokollen, wie MIP, ermöglicht wird. Die Sicherheit des SSL/TLS Handshakes, der als Basis der im Rahmen der hier entwickelten Lösung zur beidseitigen Authentifizierung eingesetzt wird, wird dort ebenfalls beschrieben. Eine zertifikatebasierte Authentifizierung verlangt eine Struktur zu Verwaltung der Schlüssel bzw. der Zertifikate, in denen die Zugehörigkeit der Schlüssel zu bestimmten Personen bescheinigt wird. Die grundlegende Funktion einer PKI wird in Kapitel 2.5 „Public Key Infrastruktur“ beschrieben. Aus den dort dargelegten Gründen wird bei der in dieser Arbeit entwickelten Lösung kein „Web of Trust“ eingesetzt.

Im vorigen Kapitel.2.7 werden schließlich andere Forschungsprojekte mit anderen Roaming Modellen und deren Probleme betrachtet. Beispielsweise ist das Problem einer Möglichkeit zur beidseitigen Authentifizierung, bei der der „Nutzer noch offline“ ist, nicht berücksichtigt und wird im Rahmen dieser Arbeit gelöst.

Nachdem in diesem Kapitel 2 die grundlegenden Technologien identifiziert und analysiert wurden, werden im nächsten Kapitel 3 Interessen der einzelnen involvierten Parteien, welche beim Vorgang des Roaming alle involviert sind, identifiziert und analysiert. Hierfür werden generische Geschäftsmodelle entwickelt, welche die Beziehungen und Interessen der beteiligten Parteien berücksichtigen.

3 Geschäftsmodelle

Ein Geschäftsmodell beschreibt, auf welche Art und Weise eine Partei Geschäfte macht, indem die Aktivitäten der involvierten Parteien erklärt werden, ihre Beziehungen untereinander beschrieben und die Möglichkeiten, Gewinne zu erwirtschaften, dargestellt werden.

Der Term Geschäftsmodell oder „Business Model“ entspricht in dieser Arbeit der Definition von Timmers: „A business model comprises an architecture for products, or service and information flows, including a description of the various business activities and their roles, a description of the potential benefits for the various business actors, and a description of the sources of revenues.” [Timm99].

Diese Geschäftsmodelle ermöglichen in dem Sinne die Entwicklung einer generischen Sicherheitsarchitektur, dass mögliche unterschiedliche Beziehungen zwischen den einzelnen relevanten im Folgenden definierten Rollen berücksichtigt werden. Dabei werden alle sinnvollen realistischen Beziehungen untersucht. Die verschiedenen untersuchten Geschäftsbeziehungen bedingen unterschiedliche Vertrauensbeziehungen zwischen den einzelnen Geschäftspartnern. Diese Vertrauensbeziehungen bilden die Grundlage für die Sicherheitsanforderungen zwischen den beteiligten Parteien.

Im nächsten Abschnitt 3.1 werden zunächst die Rollen erläutert, welche unterschiedliche Parteien oder auch Geschäftspartner annehmen können. Danach werden in den Abschnitten 3.2 bis 3.5 die Geschäftsmodelle dargestellt. Ein profundes Verständnis dieser Modelle ist notwendig, um mögliche Bedrohungen, die von betrügerischen Parteien ausgehen könnten, zu erkennen und die notwendigen technischen Lösungen zu entwickeln, welche die Interessen der ehrlichen Parteien schützen. In Abschnitt 3.6 werden die Vertrauensbeziehungen und die aus ihnen resultierenden Sicherheitsanforderungen beschrieben.

3.1 Die Rollen

Die Beschreibungen der Rollen, welche die Parteien, die den Geschäftsmodellen zugrunde liegen, annehmen können, enthalten alle relevanten Informationen, die zum Verständnis der Geschäftsmodelle gebraucht werden. Es handelt sich bei den Rollen im Einzelnen um:

- den Endverbraucher,
- den Internet Service Provider (ISP),
- das Unternehmen,
- den Zugangsnetzbetreiber,
- den PKI Service Provider und
- den Roaming Service Provider

3.1.1 Der Endverbraucher

Der Endverbraucher bzw. Nutzer ist im Sinne der hier im Folgenden betrachteten Geschäftsmodelle ein Kunde, der den “seamless security“-Dienst nutzt. Er kann dabei entweder direkt oder indirekt Kunde eines ISPs sein. Direkt bedeutet, dass er selbst persönlich einen Vertrag mir einem ISP abgeschlossen hat. Indirekt bedeutet, dass er Mitarbeiter eines Unternehmens ist, welches wiederum Kunde eines ISPs ist.

3.1.2 Der Internet Service Provider (ISP)

Der “Intenet Service Provider” – kurz ISP – stellt für Endverbraucher den Zugang zum Internet bereit, indem er ihn mit allen nötigen Ressourcen, wie z.B. Zugangs-Gateway, DNS und IP-Adresse, versorgt. Die Endverbraucher können zur Menge der eigenen Kunden des ISPs, zu denen ein vertragliches Verhältnis gegeben ist gehören oder sie können Kunden eines anderen ISPs sein, mit dem eine Vereinbarung hinsichtlich der gemeinsamen Ressourcennutzung besteht. Verträge zwischen ISPs und Kunden können genau spezifizieren was für die Nutzung welches Dienstes gilt. Die Kunden können unter anderem Eigenschaften von gebuchten Diensten wählen, wie z.B. „quality of service“ QoS oder „seamless roaming“ oder „roaming“.

Normalerweise berechnet ein ISP allen Nutzern seiner Ressourcen die Nutzung derselben. In dem hier zugrunde liegenden Modell berechnet der ISP nur seinen eigenen Vertragspartnern die Nutzung der zu Verfügung gestellten Ressourcen. Stellt ein ISP seine Ressourcen einem Kunden eines anderen ISPs zu Verfügung, stellt er dies dem anderen ISP in Rechnung. Daher muss ein ISP die relevanten Abrechnungsdaten an andere ISPs - seine Partner - übermitteln. Dies erfordert eine eindeutige Authentifizierung der Nutzer.

3.1.3 Das Unternehmen

In einigen Szenarien ist die Rolle eines Unternehmens relevant. Das Unternehmen stellt einige interne Ressourcen für seine Mitarbeiter bereit, da es ein Interesse daran hat, dass seine Mitarbeiter von überall her in der Lage sind, auf die internen Ressourcen zuzugreifen. Die Bereitstellung von korrekten und aktuellen Informationen hilft interne Unternehmensprozesse zu rationalisieren und kann helfen, die Produktivität zu erhöhen. Interne Ressourcen sollten sowohl vor unbefugtem Zugriff geschützt werden, als auch für die Mitarbeiter des entsprechenden Unternehmens leicht zugänglich sein.

Manche Unternehmen sind mittels eines ISPs an das Internet angeschlossen. Hierfür dient ein entsprechender Vertrag zwischen ISP und Unternehmen, in dem die Modalitäten festgelegt sind. Weiter gibt es Unternehmen, die ihr eigenes Internet Gateway betreiben. Diese nehmen gleichzeitig die Rolle eines ISPs für ihre Mitarbeiter an.

Des Weiteren definiert ein Unternehmen normalerweise eine „policy“, d.h. eine Strategie, welche unter Berücksichtigung der Eigenschaften der Kommunikationsbeziehungen angibt, was für die betroffenen Mitarbeiter erlaubt ist und was verboten ist. Man spricht von einer „closed policy“, wenn alles was nicht explizit erlaubt ist, verboten ist. Von einer „open policy“ spricht man, wenn alles was nicht explizit verboten ist, erlaubt ist.

Da interne Geschäftsprozesse sich in einem Unternehmen normalerweise verändern, kann sich auch die Strategie im Laufe der Zeit verändern. Die Unterstützung der Flexibilität ist bei der Verwaltung der Strategien ein oft vernachlässigter aber wichtiger Aspekt. Ein Unternehmen hat ein beträchtliches Interesse daran, dass Veränderungen der Strategien technisch so schnell wie möglich bei minimalen Kosten übernommen werden können. Dies kann ein Unternehmen vor möglichen Verlusten schützten.

3.1.4 Zugangsnetzbetreiber

Der Zugangsnetzbetreiber bietet dem mobilen Endkunden durch seine Infrastruktur die Möglichkeit, Zugang zum Netz bzw. den Diensten seines ISP zu erhalten. Es gibt im Wesentlichen zwei Arten von Zugangsnetzbetreibern, die auf Grund ihrer großen weltweiten Verbreitung relevant sind. Das sind zum einen die die Mobilfunkbetreiber, die „mobile carrier“ und zum anderen die Betreiber von WLANs. Beide werden im Folgenden beschrieben.

3.1.4.1 Mobile Carrier

Der Mobilfunkanbieter (Mobile carrier) bietet Zugang zu einem IMT-2000- oder UMTS-Netz für seine Endkunden an. Um die Dienste des mobilen Carriers zu nutzen, müssen die Endkunden, unabhängig davon ob es sich um private Nutzer oder Unternehmen handelt, einen Vertrag mit dem Mobilfunkanbieter abschließen. Hierbei kann es unterschiedliche Dienstklassen bzw. Pakte zur Auswahl geben, aus denen der Endkunde wählen kann. Entsprechend seiner Strategie kann der Mobilfunkanbieter seine Autorisierungsinformation in Abhängigkeit von dem jeweiligen Vertrag bzw. dem gewählten Paket erstellen.

Der Mobilfunkbetreiber berechnet dem Nutzer gemäß dem Vertrag die Kosten für die Nutzung der entsprechenden Ressourcen. Beides, Autorisierung und Abrechnung, benötigen die Authentifikation der Endnutzer. Im Kontext von UMTS/IMT-2000 basiert die Authentifikation der Endnutzer auf einer USIM integrated circuit card (UICC), welche ein UMTS Subscriber Identified Module (USIM) enthält.

Endnutzer, die die Dienste eines ISPs in Anspruch nehmen, können hierbei das UMTS/IMT2000-Netz eines Mobilfunkanbieters als Zugangsnetz nutzen. In speziellen Fällen spielt der Mobilfunkanbieter gleichzeitig die Rolle eines ISPs.

3.1.4.2 WLAN Provider

Ein WLAN Provider ermöglicht den Endnutzern Zugang zu seinen Ressourcen. Für gewöhnlich nutzen Endnutzer ein WLAN-Zugangsnetz, um entweder Dienste eines ISPs zu nutzen oder lokal vom Betreiber des Zugangsnetzes zu Verfügung gestellte Informationen abzurufen bzw. Dienste zu nutzen.
Ein WLAN Provider kann einen Vertrag zu einem oder mehreren ISPs haben. Wenn ein Endnutzer mit dem Zugangsnetz des WLAN Betreibers verbunden ist, kann er sofort transparent die Dienste des ISPs nutzen, mit dem der WLAN Betreiber kooperiert. In

diesem Fall nutzt der Endnutzer die Dienste des ISPs gemäß dem zwischen ISP und WLAN Provider ausgehandelten Vertrages.

Es könne zwei unterschiedliche Beziehungen unterschieden werden, die zwischen WLAN Provider und Endnutzer bestehen:

- Der Endnutzer und der WLAN Provider haben einen Vertrag abgeschlossen. In diesem Fall erhält der WLAN Provider sein Entgelt direkt vom Endnutzer. Dies kann auf Basis unterschiedlicher Modelle erfolgen. Es gibt Beispiele, bei denen ein vertragliches Verhältnis zugrunde liegt, aber auch andere Beispiele, wie zum Beispiel am Flughafen, in einem Cafe oder Hotel.
- Der Endnutzer und der WLAN Provider haben keinerlei vertragliches Verhältnis. In diesem Fall erhält der WLAN Provider sein Entgelt dafür, dass er seine Dienste anbietet, vom ISP.

3.1.5 Roaming Service Provider

Der Roaming Service Provider (RSP) stellt die Möglichkeit bereit, für den Endnutzer sich zwischen unterschiedlichen drahtlosen Netzen übergangslos zu bewegen. Solche Netze können entweder WLANs oder UMTS/IMT-2000 Netze sein. Hierbei ist das Ziel, das vereinbarte Sicherheitsniveau während des Wechselns nicht zu verändern, auch wenn zwischen den Netzen unterschiedlicher Anbieter gewechselt wird. Die Netze können auf unterschiedlichen Technologien beruhen. Es wird vorausgesetzt, dass der RSP Verträge mit ISPs und Unternehmen hat. Er hat keine direkte Beziehung zum Endnutzer, obwohl er Roaming Dienste für den Endnutzer bereitstellt.

Der RSP bietet Roaming Dienste passend zu folgenden Geschäftsmodellen an:

- ISP roaming Modell,
- Roaming mit virtual private network (VPN) Zugangsdienst - Modell,
- Seamless Roaming mit VPN Zugangsdienst Modell (mit internem HA), und
- Seamless Roaming VPN Zugangsdienst Modell (mit externem HA).

Es wird hier davon ausgegangen, dass der RSP mit einem PKI Dienstanbieter kooperiert, wie er im nächsten Abschnitt beschrieben ist.

3.1.6 PKI Service Provider

Bei einem „public key infrastructure“ (PKI) Dienstanbieter, wird im folgenden von einer Partei ausgegangen, die einen oder mehr typische PKI-Dienste anbietet, wie eine „registration authority” (RA), eine „certification authority” (CA) und die Unterstützung von Sperrlisten von Zertifikaten - CRLs genannt. Die Bereitstellung von CRL Dateien bzw. CRL Information kann via „lightweight directory access protocol“ (LDAP) server erfolgen und auch „online certificate status protocol“ (OCSP) sowie „simple certificate validation protocol“ (SCVP) Dienste beinhalten. PKI Dienste sind nötig, wenn asymmetrische Kryptographie angewandt wird. Öffentliche Schlüssel müssen authentifiziert werden, so dass sie eindeutig ihrem Besitzer zugeordnet werden können. Dies wird durch Zertifikate erreicht, welche von einer CA ausgestellt wurden.

Man kann Zertifikate zwecks Authentifikation von Endnutzern anwenden. Eine CA kann aber auch spezielle Zertifikate, so genannte Attributszertifikate, erzeugen, welche für die Autorisierung verwendet werden.

Ein Anbieter von PKI Diensten kann von einer bestimmten Partei betrieben werden, wie z.B. einer der oben genannten: Unternehmen oder RSP. Zertifikaten sollte nur vertraut werden, wenn sie gültig sind und der entsprechende Aussteller aus Sicht des Überprüfenden vertrauenswürdig ist. Dementsprechend sollte einem Zertifikat nur vertraut werden, wenn ein Pfad vom Vertrauensanker des Überprüfers zum Aussteller des Zertifikates existiert, wie z.B. in dem einfachen Fall, dass der Überprüfer den Aussteller persönlich kennt und ihm vertraut.

PKI Dienste, die von Unternehmen für ihre Mitarbeiter angeboten werden, können den Nachteil haben, dass das Finden von Validierungspfaden von einer gegebenen Menge an Vertrauensankern schwierig oder sogar unmöglich sein kann, da diese Dienste typischerweise nicht sehr gut angebunden sind. In solchen Fällen kann eine Partei, welche ein solches Zertifikat benutzt nicht authentifiziert werden. Zusätzlich ist die Autorisierungsüberprüfung unter solchen Bedingungen auch noch unmöglich, falls die Autorisierung auf Attributszertifikaten basiert.

3.2 ISP Roaming Geschäftsmodell

Im Rahmen des ISP Roaming Modells geht es darum, unterschiedlichen Nutzern durch einen “Roaming” Dienst den Internetzugang über unterschiedliche ISPs zu ermöglichen. In diesem Zusammenhang sind Endnutzer und ISP einander möglicherweise bekannt.

3.2.1 Beschreibung der Beziehungen

Im Rahmen dieses Geschäftsmodells müssen die Beziehungen zwischen folgenden Rollen beachtet werden: Endnutzer, Zugangsnetzbetreiber, kontaktierter ISP, RSP, Heim-ISP und PKI Dienstanbieter. Diese stehen jeweils paarweise, wie folgt, in unterschiedliche Beziehung zueinander:
Der Endnutzer und der ISP unterzeichnen einen Vertrag. In diesem ist u. a. festgelegt, dass der Endnutzer an den ISP Geld bezahlt dafür, dass er die Dienste des ISPs nutzt.

- Der ISP kooperiert mit dem RSP auf Basis eines Vertrages, um seinen Kunden „Roaming Dienste“ anzubieten.
- Bezüglich des Austauschs von Geld unter den ISPs, ist der RSP dafür verantwortlich eine Menge von Regeln, die für alle ISPs gelten zu definieren und durchzusetzen: Der Heim-ISP stellt die für einen Endnutzer gebrachten Leistungen in Rechnung und leitet gegebenenfalls dieses Geld weiter zum kontaktierten ISP. Dies kann direkt oder über den RSP erfolgen.
- Die ISPs stellen Beschreibung von Strategien bereit, was ihre Kunden dürfen und was nicht. Der kontaktierte ISP stellt Dienste bereit passend zu der beschriebenen Strategie, die vom Heim-ISP bereitgestellt wird. Hierbei kann der RSP involviert werden, wenn die Strategie in der Datenbank des RSP gespeichert wurde oder der

RSP die Zertifikate oder Attributszertifikate, welche Strategiebeschreibungen beinhalten, ausgestellt hat.
- Der Zugangsnetzbetreiber hat einen Vertrag mit einem ISP. Dieser ISP wird hier i. d. R. als kontaktierter ISP bezeichnet.
- Der Zugangsnetzbetreiber stellt u. a. Zugang zum Internet über sein Netz bereit. Er kann hierfür einen Vertrag mit dem Nutzer haben oder nicht.
- Der RSP kooperiert mit dem PKI Dienstanbieter im dem Falle, dass er diesen Dienst nicht selber anbietet.

3.2.2 Aktivitäten

Die Hauptaktivitäten, der in das hier zugrunde gelegte Geschäftsmodell involvierten Rollen, werden nun beschrieben. Da die Rolle des Unternehmens in diesem Modell nicht existent ist, wird davon ausgegangen, dass die CA entweder vom RSP betrieben wird oder mit dem RSP sehr eng zusammenarbeitet. Die Aktivitäten sind im Einzelnen:

- Der RSP ist verantwortlich für die Ausstellung der Endnutzer Zertifikate. Es wird hier weiterhin vorausgesetzt, dass die Endnutzerzertifikate eine Referenz vom RSP auf den entsprechenden ISP haben, mit dem die Nutzer ein vertragliches Verhältnis eingegangen sind, damit die Nutzer einem ISP zugeordnet werden können.
- Der ISP betreibt die Registration Authority (RA).
- Der Endnutzer erhält sein Zertifikat von seinem ISP. Der ISP könnte auch gleichzeitig der Mobilfunkbetreiber sein.
- Der Endnutzer muss in der Lage sein, die Signaturen des ISPs zu überprüfen, welche vom RSP ausgestellt wurden. In dem Fall, dass er nicht in der Lage ist die Signatur zu verifizieren, muss der Endnutzer dem kontaktierten ISP bezüglich der Korrektheit der signierten Daten vertrauen. Wenn er dieses vertrauen nicht hat, kann er den Prozess abbrechen.
- Die Endnutzerauthentifikation wird normalerweise vom kontaktierten ISP durchgeführt. Wenn die Authentifikation durch den kontaktierten ISP nicht möglich ist, dann wird sie zum RSP weiterdelegiert. Die Delegation der Authentifizierung kann nötig werden, wenn der kontaktierte ISP nicht in der Lage ist, den korrekten Algorithmus für die Verifikation oder der kontaktierte ISP nicht in der Lage ist, einen gegebenen Schlüssel anzuwenden auf Grund von Längenregulierungen.
- Wenn die Authentifizierung zum RSP delegiert wird, wird sie vom RSP nicht mehr weiter delegiert.
- Zugangskontrollentscheidungen für Dienstenutzung werden vom kontaktierten ISP durchgesetzt in Abhängigkeit der ihm vom Heim-ISP gegebenen Information zur Autorisierung.
- Im Falle einer auf Zertifikatserweiterungen basierenden Autorisierung kann die Zugangskontrollentscheidung vom kontaktierten ISP auf Basis der Information im Zertifikat selbst getroffen werden, falls keine Delegation notwendig ist.
- Im dem Fall, dass die Autorisierungsinformation ausschließlich in Datenbanken gespeichert ist, sendet der ISP eine Anforderung (Request) zum RSP um die

entsprechende Autorisierungsinformation, welche für die Durchsetzung erforderlich ist, zu erhalten.

- Der kontaktierte ISP sammelt Abrechnungsinformation für die Dienste, welche vom Endnutzer in Anspruch genommen wurden und leitet diese Information weiter zum Heim-ISP. Dies kann über den RSP erfolgen oder auf direktem Wege.
- Der Heim-ISP stellt dem Endnutzer eine Rechnung gemäß der gesammelten Abrechnungsinformation und transferiert das Geld zum kontaktierten ISP gemäß dem zugrunde liegenden Vertrag.
- Der Endnutzer kann auch an den Zugangsnetzbetreiber zahlen und der Zugangsnetzbetreiber kann die ISPs zahlen müssen.

3.2.3 Erlöse

Zusätzliche Informationen bezüglich der möglichen Flüsse der Erlöse lassen es zu, eine Sicherheitsarchitektur zu entwickeln, bei der auch die Abrechnung berücksichtigt werden kann. Der Austausch von die Abrechnung betreffenden Informationen wird von denjenigen Parteien initiiert, welche den Endnutzern Dienste bereitgestellt haben, und dementsprechend als Folge einen monetären Gegenwert erwarten können. Da man nicht voraussetzen kann, dass alle Parteien, die in den Austausch von Geldwerten involviert sind, einander bedingungslos vertrauen, ist es notwendig Sicherheitsmechanismen einzusetzen, die die Interessen der ehrlichen Parteien schützen.
Im Folgenden werden hier die potentiellen Geldflüsse dargestellt, wobei die folgenden grundlegenden Annahmen getroffen werden:

- Der Heim-ISP erhält Geld vom Endnutzer
- Der Heim-ISP zahlt Geld an den RSP für seinen Roaming Dienst
- Die ISPs tauschen Geld über den RSP aus
- Der Zugangsnetzbetreiber erhält entweder Geld
    - vom ISP oder
    - vom Endnutzer

Die Abbildung 29 zeigt Geldflüsse, wie sie typisch sind in einem realistischen Roaming Szenario. Der Nutzer zahlt an den ISP, mit dem er einen Vertrag hat, für die Inanspruchnahme seiner Dienste. Der Zugangsnetzbetreiber, z.B. ein Internetcafe hat ebenfalls einen Vertrag mit einem ISP und zahlt an diesen ISP den entsprechenden vertraglich geregelten Betrag. Zusätzlich zahlt der Endnutzer an den Zugangsnetzbetreiber dafür, dass er ins Internet gelangen kann. Der Internet Dienstanbieter zahlt wiederum an den RSP dafür, dass seine Nutzer „Roamen“ können.

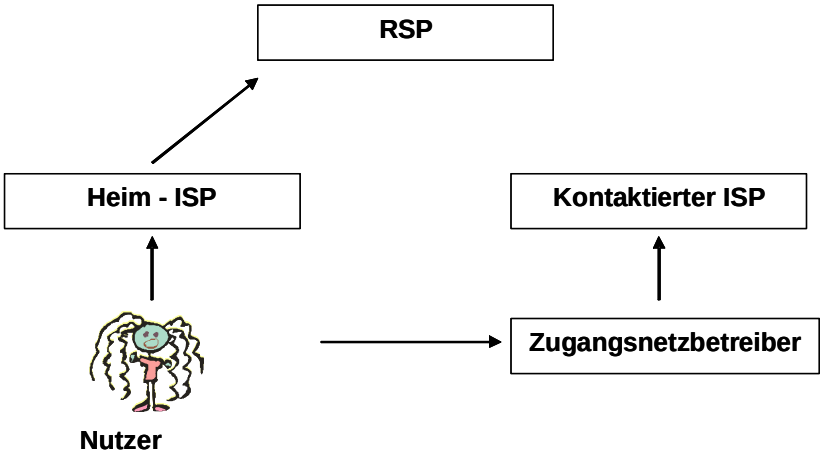


Abbildung 29: Geldfluss A

In Abbildung 30 ist eine Alternative Möglichkeit für die Geldflüsse dargestellt. Auch hier zahlt der Nutzer an den ISP, mit dem er einen Vertrag hat, für die Inanspruchnahme seiner Dienste, und der Internet Dienstanbieter zahlt wiederum an den RSP dafür, dass seine Nutzer „Roamen“ können. Allerdings zahlt der Heim-ISP an den kontaktierten ISP dafür Geld, dass dieser seinem Kunden – dem Endnutzer – den Zugang zum Internet ermöglicht. Der kontaktierte ISP wiederum zahlt an den Zugangsnetzbetreiber dafür, dass er den Endnutzer zu ihm weiterleitet, um Internetzugang zu erhalten.

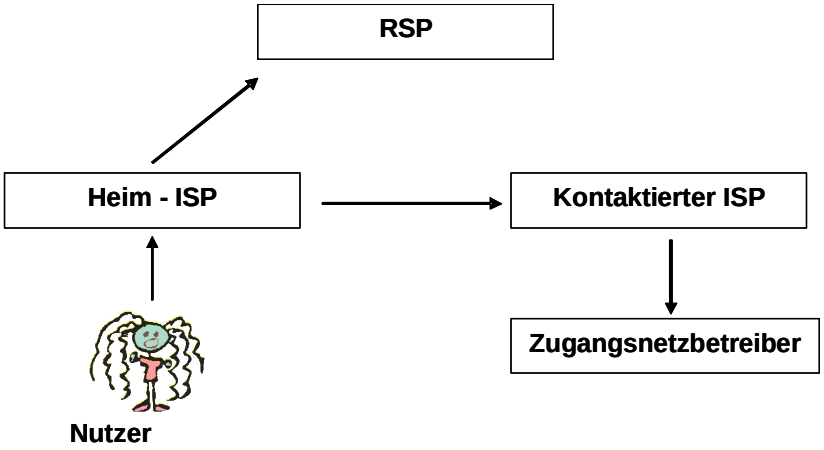


Abbildung 30: Geldfluss B

In Abbildung 31 ist eine weitere Alternative aufgezeigt. Die hier dargestellten Flüsse entsprechen denen der vorigen Abbildung mit folgendem Unterschied: Der Heim-ISP zahlt jetzt nicht mehr direkt an den kontaktierten ISP dafür, dass dieser seinen Kunden seine Dienste anbietet, sondern er bezahlt den entsprechenden Betrag an den RSP. Dieser wiederum zahlt den vereinbarten Betrag an den kontaktierten ISP. Der RSP hat mit beiden ISPs eine entsprechende vertragliche Vereinbarung.

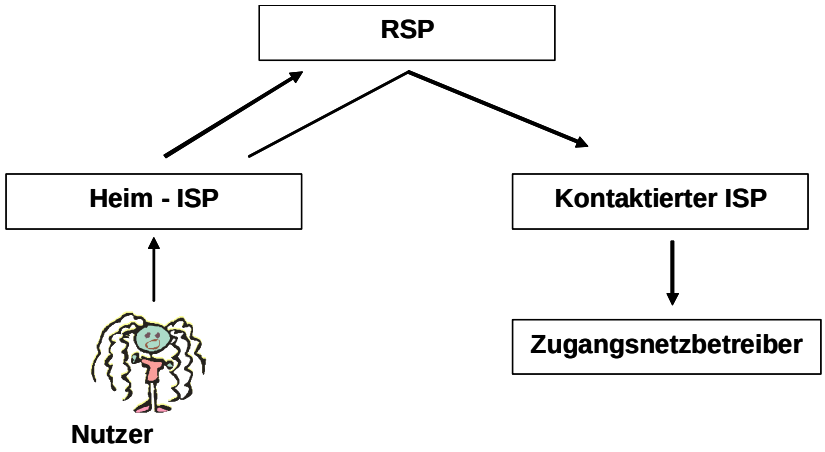


Abbildung 31: Geldfluss C

Der in Abbildung 32 dargestellte Geldfluss D entspricht dem Fluss C aus der vorigen Abbildung nur mit dem Unterschied, dass nicht der kontaktierte ISP an den Zugangsnetzbetreiber Geld zahlt, sondern wie in Geldfluss A der Endnutzer direkt den Zugangsnetzbetreiber für seine Dienste entlohnt.

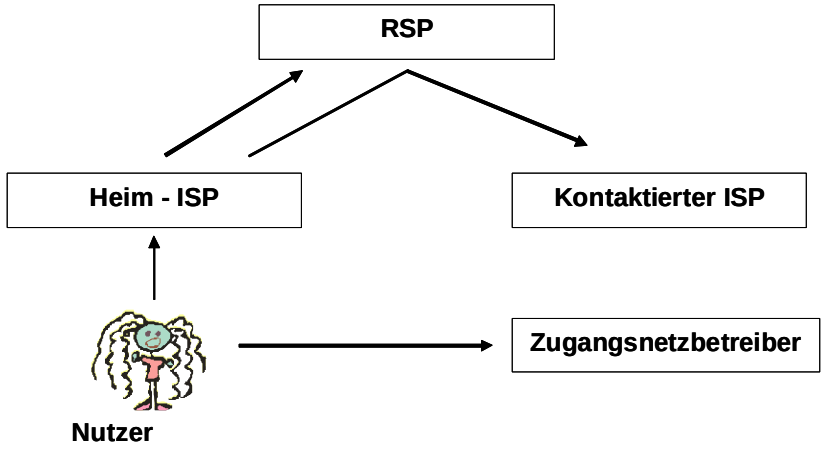


Abbildung 32: Geldfluss D

Eine weitere Alternative befindet sich in Abbildung 33 Der hier dargestellte Geldfluss E entspricht dem Geldfluss A abgesehen davon, dass nicht der Zugangsnetzbetreiber an den kontaktierten ISP Geld zahlt, sondern wie bei Geldfluss B der Heim-ISP den kontaktierten ISP dafür bezahlt, dass er seinen Kunden den Internetzugang ermöglicht.

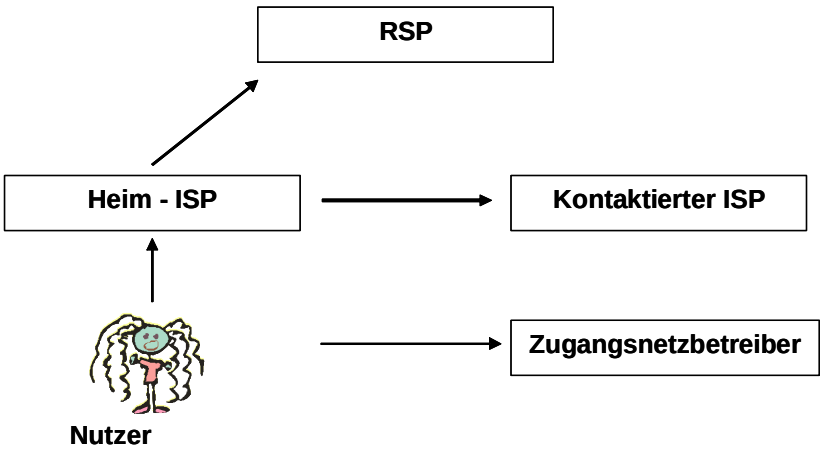


Abbildung 33: Geldfluss E

Andere Möglichkeiten sind in der Realität nicht sinnvoll, solange die Leistungserbringer von den Leistungsnutzern bezahlt werden.

3.2.4 Vorteile

Die Vorteile, welche die einzelnen Rollen von dem hier entwickelten Geschäftsmodell haben, werden im Folgenden skizziert.

3.2.4.1 Vorteile für RSP

Der RSP hat die Möglichkeit, Einkommen für das Anbieten seines Dienstes zu generieren.

3.2.4.2 Vorteile für Endnutzer

Folgende Vorteile ergeben sich für den Endnutzer:

- Die Nutzbarkeit steigt durch die Mobilitätsunterstützung.
- Es besteht keine Notwendigkeit, mehrere Vertragsverhältnisse einzugehen.
- Dem Nutzer wird nur eine Rechnung gestellt und er muss nur eine Zahlung zu einem Partner durchführen.

3.2.4.3 Vorteile für ISPs

Folgende Vorteile ergeben sich für dis ISPs:

- Die Anzahl der Kooperationen, in die ISPs involviert sind, lässt sich von n(n-1)/2 auf n im Falle von n ISPs reduzieren.
- Es können mehr Endnutzer erreicht werden ohne mehr Vertragsverhältnisse eingehen zu müssen. Die ist ein Wettbewerbsvorteil in dem Sinne, dass die Attraktivität durch die das „Roaming“ Angebot steigt, wodurch mehr Kunden gewonnen werden können.
- Es bietet sich die Möglichkeit eines Wertetransfers zwischen ISPs, die keine Vertragspartner sind, die sich gar nicht kennen.

3.2.4.4 Vorteile für Zugangsnetzbetreiber

Der Zugangsnetzbetreiber hat die Möglichkeit, Einkommen für die Bereitstellung seines Dienstes zu generieren.

3.3 Roaming VPN Zugangsdienst Geschäftsmodell

Das Roaming VPN Zugangsdienst Modell stellt sicheres Roaming bereit, um Endnutzern zu erlauben, zwischen unterschiedlichen ISPs einen Internetzugang nutzen und um eine VPN Verbindung zu einem sicheren Netz herstellen zu können. Des Weiteren kann der Endnutzer in der Art eingeschränkt werden, dass er nur Verbindung zu einer gegebenen Adresse herstellen kann. Zusätzlich wird die Etablierung einer VPN Verbindung zwischen End-Nutzer und Unternehmen unterstützt. Die Verwendung der sicheren VPN Verbindung kann sogar erzwungen werden. Zusätzlich zu den im ISP Roaming Modell vertretenen Rollen kommt hier die Rolle des Unternehmens hinzu. Die PKI spezifischen Dienste können auch von Komponenten, welche vom Unternehmen verwaltet und betrieben werden, angeboten werden.
Bezüglich der Partei, die die VPN Schlüssel verwaltet, kann man zwei Fälle unterscheiden:

- Der RSP authentifiziert den Endnutzer und stellt die VPN Schlüssel bereit. Hierbei hat der RSP jederzeit die Möglichkeit hat, die gesamte Kommunikation zu belauschen und zu manipulieren. Andererseits hat dies den Vorteil, dass die Geschwindigkeit des Prozesses höher ist, als im zweiten Fall.
- Der RSP authentifiziert den Endnutzer, stellt aber die VPN Schlüssel nicht bereit. Der Endnutzer und das Unternehmen generieren ihre VPN Schlüssel selber.

Der Fokus liegt hier auf dem Fall, in dem der RSP die Schlüssel verteilt. Dieser Fall ist deshalb sinnvoll, weil dadurch der Aufbau einer vertraulichen Verbindung schneller erfolgen kann, als wenn Endnutzer und Unternehmen z. B. eines der in Kapitel 2.3.2.3 beschriebenen Protokolle miteinander abwickeln. Der symmetrische Schlüssel sollte jeweils mit dem öffentlichen Schlüssel der beiden Kommunikationspartner verschlüsselt an dieselben geschickt werden. Der Fall, in dem der Endnutzer und das Unternehmen die Schlüssel auf ihre Weise generieren, wird hier nicht weiter betrachtet, da die Generierung und der Austausch von Schlüsseln in diversen Standardprotokollen, wie z.B. SSL hinreichend beschrieben sind.

3.3.1 Beschreibung der Beziehungen

Im Rahmen des Roaming VPN Zugangsdienst Modells müssen wieder die Beziehungen zwischen den Rollen Endnutzer, Zugangsnetzbetreiber, kontaktierter ISP, RSP, Heim-ISP, PKI Dienstanbieter und zusätzlich der Rolle des Unternehmens betrachtet werden. Diese stehen jeweils paarweise in unterschiedlichen Beziehung zueinander, wie folgt:

- Das Unternehmen und der ISP unterzeichnen Verträge, Das Unternehmen bezahlt dabei Geld an den ISP für die Nutzung der vom ISP angebotenen Dienste.
- Das Unternehmen und der RSP schließen einen Vertrag dahingehend ab, dass der RSP Zertifikate der Mitarbeiter des Unternehmens akzeptiert, auch wenn sie von einer unternehmenseigenen CA ausgestellt werden.

- Der ISP kooperiert mit dem RSP auf vertraglicher Basis, um seinen Kunden einen Roaming Dienst anzubieten.
- Für die Abrechnungen zwischen den ISPs definiert der RSP Regeln, nach denen die ISPs sich zu richten haben. Der Heim-ISP stellt die Rechnungen an seine Kunden und gibt das Geld zum kontaktierten ISP entweder direkt oder über den RSP weiter.
- Das Unternehmen stellt dem RSP die Beschreibung seiner Sicherheitsstrategie zu Verfügung.
- Die ISPs stellen Beschreibungen ihrer Strategien für einander bereit. Der kontaktierte ISP bietet dabei entsprechend der Strategie des Heim-ISPs Dienste an. Der RSP kann hier involviert werden, wenn die Strategie des Heim-ISPs bei ihm gespeichert ist oder der RSP Zertifikate oder Attributzertifikate ausstellt, welche die Beschreibungen der Strategien enthalten.
- Der Zugangsnetzbetreiber hat einen Vertrag mit dem kontaktierten ISP.
- Der Zugangsnetzbetreiber stellt seinen Zugangspunkt dem Endnutzer entweder mit oder ohne Vertrag zu Verfügung.
- Der RSP kann mit einem PKI Betreiber zusammenarbeiten, wenn er die entsprechenden Dienste nicht selbst anbietet.
- In dem Fall, dass das Unternehmen seine eigene CA betreibt, akzeptiert der RSP die Zertifikate, welche von der CA ausgestellt wurden

3.3.2 Aktivitäten

Die Hauptaktivitäten, der in das hier zugrunde gelegte Geschäftsmodell involvierten Rollen, werden nun beschrieben. Da die Rolle des Unternehmen in diesem Modell existent ist, kann die CA entweder vom Unternehmen oder vom RSP oder von einem Dritten, der mit einem dieser beiden eng zusammenarbeitet betrieben werden. Die Aktivitäten sind im einzelnen:

- Ein Unternehmen kann eine Unternehmens CA betreiben und noch zusätzliche PKI bezogene Dienste, wie z.B. CRLs anbieten.
- Der RSP ist dafür verantwortlich, Zertifikate für Endnutzer auszustellen. Er betreibt seine eigene CA oder nutzt die Dienste einer dritten Partei.
- Der Endnutzer erhält seine Zertifikate über sein Unternehmen entweder von der Unternehmenseigenen CA oder von der CA des seinem Unternehmen assoziierten RSPs.
- Der Endnutzer sollte in der Lage sein, Signaturen von ISPs zu überprüfen, die auf Zertifikaten beruhen, die vom RSP ausgestellt wurden. Ist der Endnutzer nicht in der Lage, die Signaturüberprüfung durchzuführen, muss er dem ISP hinsichtlich der Korrektheit der signierten Daten vertrauen. Falls kein Vertrauen gegeben ist, kann der Endnutzer den Prozess abbrechen.
- Der RSP generiert die VPN Sitzungsschlüssel für die Kommunikation zwischen Unternehmen und Endnutzer.
- Die Authentifizierung des Endnutzers wird vom kontaktierten ISP durchgeführt. Wenn die Authentifizierung nicht möglich ist, wird sie an den RSP delegiert. Die Delegation der Authentifizierung ist z.B. notwendig, wenn der kontaktierte ISP nicht in der Lage ist, einen bei der Authentifizierung erforderlichen Algorithmus

auszuführen oder auf Grund von staatlichen Regulierungen die gegebenen Schlüssel wegen ihrer Länge nicht einsetzen darf.

- Der RSP authentifiziert den Nutzer, wenn er ein von einem Unternehmen ausgestelltes Zertifikat hat, welches dem kontaktierten ISP unbekannt ist. Im Allgemeinen kann man nicht davon ausgehen, dass ein kontaktierter ISP einen Validierungspfad zu einer unternehmenseigenen CA herstellen kann.
- Ein kontaktierter ISP entscheidet prinzipiell über den Zugriff auf seine Dienste unabhängig von der Autorisierungsinformation. Ob ein Endnutzer einen Dienst in Anspruch nehmen darf oder nicht hängt, jedoch von der entsprechenden Autorisierungsinformation ab. Die Autorisierungsinformation wird entweder vom Unternehmen, vom Heim-ISP oder vom RSP zu Verfügung gestellt.
- Im Falle von auf Zertifikatserweiterungen basierender Autorisierung, wie unten in Kapitel 7 beschrieben, können die Zugangskontrollentscheidungen vom kontaktierten ISP selbst getroffen werden.
- In dem Fall, dass die Autorisierungsinformation nur in Datenbanken gespeichert ist, sendet der kontaktierte ISP eine Anfrage zum RSP bezüglich der Zugangskontrollentscheidungen, welche vom kontaktierten ISP umzusetzen sind.
- Der kontaktierte ISP sammelt Abrechnungsinformation für vom Endnutzer in Anspruch genommene Dienste und leitet diese Information zum Heim-ISP weiter.
- Der Heim-ISP stellt dem Unternehmen gemäß den gesammelten Abrechnungsinformationen eine Rechnung und transferiert einen entsprechenden Anteil zum kontaktierten ISP entsprechend dem zugrunde liegenden Vertrag. Das Geld kann hierbei über den RSP fließen.

3.3.3 Erlöse

Um Sicherheitsanforderungen für die Abrechnung zu entwickeln, ist es hilfreich die möglichen Geldflüsse zu analysieren. Damit können Sicherheitsanforderungen aufgestellt werden und eine Lösung entwickelt werden, die der Abrechnung gerecht werden kann.

Die Geldflüsse entsprechen denen des ISP Roaming Modells aus Kapitel 3.2.3 und werden hier deshalb nicht nochmals dargestellt. Der einzige Unterschied besteht darin, dass der Endnutzer als Mitarbeiter eines Unternehmens nicht selber die in Anspruch genommenen Dienste bezahlt, sondern das Unternehmen.

3.3.4 Vorteile des Modells

Die Vorteile, welche die einzelnen Rollen von dem hier entwickelten Geschäftsmodell haben, werden im Folgenden skizziert.

3.3.4.1 Vorteile für den RSP

Der RSP hat die Möglichkeit, Einkommen für das Anbieten seines Dienstes zu generieren.

3.3.4.2 Vorteile für Unternehmen und Endnutzer

Folgende Vorteile ergeben sich für das Unternehmen und den Endnutzer:

- Die Nutzbarkeit steigt durch die Mobilitätsunterstützung.
- Es besteht keine Notwendigkeit, mehrere Vertragsverhältnisse mit verschiedenen ISPs einzugehen.
- Dem Unternehmen bzw. Nutzer wird nur eine Rechnung gestellt und er muss nur eine Zahlung zu einem Partner durchführen.
- Der Aufbau einer VPN Verbindung zum Unternehmen durch die Mitarbeiter ist möglich und kann sogar erzwungen werden

3.3.4.3 Vorteile für ISPs

Folgende Vorteile ergeben sich für dis ISPs:

- Die Anzahl der Kooperationen, in die ISPs involviert sind, lässt sich von n(n-1)/2 auf n im Falle von n ISPs reduzieren.
- Es können mehr Endnutzer erreicht werden ohne mehr Vertragsverhältnisse eingehen zu müssen. Die ist ein Wettbewerbsvorteil in dem Sinne, dass die Attraktivität durch die das „Roaming“ Angebot steigt, wodurch mehr Kunden gewonnen werden können.
- Die zusätzlich zum ISP Roaming Modell gegebene Möglichkeit für Kunden des ISPs ein VPN aufzubauen, erhöht die Attraktivität des ISPs. Damit eröffnet sich eine weitere Einnahmequelle für den ISP.
- Es bietet sich die Möglichkeit eines Wertetransfers zwischen ISPs, die keine Vertragspartner sind, die sich gar nicht kennen.

3.3.4.4 Vorteile für Zugangsnetzbetreiber

Der Zugangsnetzbetreiber hat die Möglichkeit, Einkommen für die Bereitstellung seines Dienstes zu generieren.

3.4 Seamless Roaming VPN Zugangsdienst Modell mit internem HA

Das Seamless Roaming VPN Zugangsdienst Geschäftsmodell stellt dem Endnutzer “seamless secure roaming” bereit mit dem Ziel, dem Endnutzer einen Internetzugang über unterschiedliche ISPs zu ermöglichen. Dabei kann der Endnutzer so eingeschränkt werden, dass er Verbindungen nur zu einem bestimmten Ziel etablieren kann, wie z.B. einem Unternehmen. Zusätzlich wird die Einrichtung einer mobilen IP - VPN Verbindung zwischen Endnutzer und Unternehmen unterstützt.

Wie schon beim Roaming VPN Zugangsdienst Modell, kann man bezüglich der Partei, die die VPN Schlüssel verwaltet, zwei Fälle unterscheiden:

- Der RSP authentifiziert den Endnutzer und stellt die VPN Schlüssel bereit. Hierbei hat der RSP jederzeit die Möglichkeit, die gesamte Kommunikation zu belauschen und zu manipulieren. Andererseits hat dies den Vorteil, dass die Geschwindigkeit des Prozesses höher ist, als im zweiten Fall.

- Der RSP authentifiziert den Endnutzer, stellt aber die VPN Schlüssel nicht bereit. Der Endnutzer und das Unternehmen generieren ihre VPN Schlüssel selber.

Der HA bereitet dem Endnutzer die Möglichkeit übergangslos zwischen Netzen zu wechseln. Dies bedeutet im Wesentlichen folgendes:

- Der Endnutzer kann, während er umherwandert, kontaktiert werden.
- Der Endnutzer kann, während er verbunden ist, seinen Ort wechseln.
- Der Endnutzer kann die Zugangstechnologie wechseln, während er verbunden ist.

3.4.1 Beschreibung der Beziehungen

Im Rahmen dieses Modells müssen die Beziehungen zwischen folgenden Rollen beachtet werden: Endnutzer, Zugangsnetzbetreiber, kontaktierter ISP, RSP, Heim-ISP, Unternehmen und PKI Dienstanbieter. Diese stehen jeweils paarweise, wie folgt, in unterschiedlicher Beziehung zueinander. Diese entsprechen im Wesentlichen den Beziehungen des Roaming VPN Zugangsdienst Geschäftsmodells:

- Das Unternehmen und der ISP unterzeichnen Verträge, Das Unternehmen bezahlt dabei Geld an den ISP für die Nutzung der vom ISP angebotenen Dienste.
- Das Unternehmen und der RSP schließen einen Vertrag dahingehend ab, dass der RSP Zertifikate der Mitarbeiter des Unternehmens akzeptiert, auch wenn sie von einer unternehmenseigenen CA ausgestellt werden.
- Der ISP kooperiert mit dem RSP auf vertraglicher Basis, um seinen Kunden einen Roaming Dienst anzubieten.
- Für die Abrechnungen zwischen den ISPs definiert der RSP Regeln, nach denen die ISPs sich zu richten haben. Der Heim-ISP stellt die Rechnungen an seine Kunden und gibt das Geld zum kontaktierten ISP entweder direkt oder über den RSP weiter.
- Das Unternehmen stellt dem RSP die Beschreibung seiner Sicherheitsstrategie zu Verfügung.
- Die ISPs stellen Beschreibungen ihrer Strategien für einander bereit. Der kontaktierte ISP bietet dabei entsprechend der Strategie des Heim-ISPs Dienste an. Der RSP kann hier involviert werden, wenn die Strategie des Heim-ISPs bei ihm gespeichert ist oder der RSP Zertifikate oder Attributzertifikate ausstellt, welche die Beschreibungen der Strategien enthalten.
- Der Zugangsnetzbetreiber hat einen Vertrag mit dem kontaktierten ISP.
- Der Zugangsnetzbetreiber stellt seinen Zugangspunkt dem Endnutzer entweder mit oder ohne Vertrag zu Verfügung.
- Der RSP kann mit einem PKI Betreiber zusammenarbeiten, wenn er die entsprechenden Dienste nicht selbst anbietet.
- In dem Fall, dass das Unternehmen seine eigene CA betreibt, akzeptiert der RSP die Zertifikate, welche von der CA ausgestellt wurden.

3.4.2 Aktivitäten

Die Hauptaktivitäten, der in das hier zugrunde gelegte Geschäftsmodell involvierten Rollen, werden nun beschrieben. Da die Rolle des Unternehmen in diesem Modell

existent ist, kann die CA entweder vom Unternehmen oder vom RSP betrieben werden oder von einem Dritten, der mit einem dieser beiden eng zusammenarbeitet. Die Aktivitäten sind im Einzelnen:

- Unternehmen können eine Unternehmens CA betreiben und noch zusätzliche PKI bezogene Dienste, wie z.B. CRLs anbieten.
- Der RSP ist dafür verantwortlich, Zertifikate für Endnutzer auszustellen, der RSP betreibt seine eigene CA oder nutzt Dienste einer dritten Partei.
- Der Endnutzer erhält seine Zertifikate über sein Unternehmen entweder von der Unternehmenseigenen CA oder von der CA des seinem Unternehmen assoziierten RSP
- Das Unternehmen verwaltet bzw. betreibt den HA des Endnutzers.
- Der Endnutzer sollte in der Lage sein, Signaturen von ISPs zu überprüfen, die auf Zertifikaten beruhen, die vom RSP ausgestellt wurden. In dem Fall, dass der Endnutzer nicht in der Lage ist, die Signaturüberprüfung durchzuführen, muss der Endnutzer dem ISP der Korrektheit der signierten Daten vertrauen. Falls kein Vertrauen gegeben ist, kann der Endnutzer den Prozess abbrechen.
- Der RSP generiert die VPN Sitzungsschlüssel für die Kommunikation zwischen den Unternehmen und dem Endnutzer.
- Die Authentifizierung des Endnutzers wird vom kontaktierten ISP durchgeführt. Wenn die Authentifizierung nicht möglich ist, wird sie an den RSP delegiert. Die Delegation der Authentifizierung ist z.B. notwendig, wenn der kontaktierte ISP nicht in der Lage ist, einen bei der Authentifizierung erforderlichen Algorithmus auszuführen oder auf Grund von staatlichen Regulierungen die gegebenen Schlüssel wegen ihrer Länge nicht einsetzen darf.
- Der RSP authentifiziert den Nutzer, wenn er ein von einem Unternehmen ausgestelltes Zertifikat hat, welches dem kontaktierten ISP unbekannt ist. Im Allgemeinen kann man nicht davon ausgehen, dass ein kontaktierter ISP einen Validierungspfad zu einer unternehmenseigenen CA herstellen kann.
- Ein kontaktierter ISP entscheidet prinzipiell über den Zugriff auf seine Dienste unabhängig von der Autorisierungsinformation. Ob ein Endnutzer einen Dienst in Anspruch nehmen darf oder nicht hängt, jedoch von der entsprechenden Autorisierungsinformation ab. Die Autorisierungsinformation wird entweder vom Unternehmen, vom Heim-ISP oder vom RSP zu Verfügung gestellt.
- Im Falle von auf Zertifikatserweiterungen basierender Autorisierung, wie unten in Kapitel 7 beschrieben, können die Zugangskontrollentscheidungen vom kontaktierten ISP selbst getroffen werden.
- In dem Fall, dass die Autorisierungsinformation nur in Datenbanken gespeichert ist, sendet der kontaktierte ISP eine Anfrage zum RSP bezüglich der Zugangskontrollentscheidungen, welche vom kontaktierten ISP umzusetzen sind.
- Der kontaktierte ISP sammelt Abrechnungsinformation für vom Endnutzer in Anspruch genommene Dienste und leitet diese Information zum Heim-ISP weiter.
- Der Heim-ISP stellt dem Unternehmen passend zu der gesammelten Abrechnungsinformation eine Rechnung und transferiert einen entsprechenden

Anteil zum kontaktierten ISP entsprechend dem zugrunde liegenden Vertrag. Das Geld kann hierbei über den RSP fließen oder nicht.

3.4.3 Erlöse

Das in Abschnitt 3.3.3 bereits gesagte gilt auch hier.

3.4.4 Vorteile

Die Vorteile, welche die einzelnen Rollen von dem hier entwickelten Geschäftsmodell haben, werden im Folgenden skizziert.

3.4.4.1 Vorteil für den RSP

Der RSP hat die Möglichkeit, Einkommen für das Anbieten seines Dienstes zu generieren.

3.4.4.2 Vorteil für Unternehmen / Endnutzer

Folgende Vorteile ergeben sich für das Unternehmen und den Endnutzer:

- Die Nutzbarkeit steigt durch die Mobilitätsunterstützung.
- Es besteht keine Notwendigkeit, mehrere Vertragsverhältnisse mit verschiedenen ISPs einzugehen.
- Dem Unternehmen bzw. Nutzer wird nur eine Rechnung gestellt und er muss nur eine Zahlung zu einem Partner durchführen.
- Der Aufbau einer VPN Verbindung zum Unternehmen durch die Mitarbeiter ist möglich und kann sogar erzwungen werden
- Mobilitätsunterstützung während der Kommuniktaion durch übergangsloses Roaming
- Unterstützung von Flexibilität bezüglich der dem Zugangsnetz zugrunde liegenden Technologie.

3.4.4.3 Vorteil für die ISPs

Folgende Vorteile ergeben sich für dis ISPs:

- Die Anzahl der Kooperationen, in die ISPs involviert sind, lässt sich von n(n-1)/2 auf n im Falle von n ISPs reduzieren.
- Es können mehr Endnutzer erreicht werden ohne mehr Vertragsverhältnisse eingehen zu müssen. Die ist ein Wettbewerbsvorteil in dem Sinne, dass die Attraktivität durch die das „Roaming“ Angebot steigt, wodurch mehr Kunden gewonnen werden können.
- Die zusätzlich zum ISP Roaming Modell gegebene Möglichkeit für Kunden des ISPs ein VPN aufzubauen, erhöht die Attraktivität des ISPs. Damit eröffnet sich eine weitere Einnahmequelle für den ISP.
- Es bietet sich die Möglichkeit eines Wertetransfers zwischen ISPs, die keine Vertragspartner sind, die sich gar nicht kennen.

3.4.4.4 Vorteil für Zugangsnetzbetreiber

Der Zugangsnetzbetreiber hat die Möglichkeit, Einkommen für die Bereitstellung seines Dienstes zu generieren.

3.5 Seamless Roaming VPN Zugangsdienst Modell mit externem HA

Das Seamless Roaming VPN Zugangsdienst Geschäftsmodell mit externem HA stellt dem Endnutzer "seamless secure roaming" bereit mit dem Ziel, dem Endnutzer einen Internetzugang über unterschiedliche ISPs zu ermöglichen. Der Unterschied zum vorher dargestellten Modell ist, dass der Home Agent (HA) sich beim RSP befindet. Dieser Ansatz könnte für kleinere Unternehmen interessant sein.
Der RSP authentifiziert den Endnutzer und das Unternehmen und verteilt dann an beide die nötigen VPN Schlüssel. Es stellt ein grundsätzliches Sicherheitsproblem dar, wenn der RSP die Schlüssel kennt. Dafür ist dieses Modell vorteilhafter bezüglich der Performance. Im Unterscheid zum VPN Modell mit internem HA ist hier der RSP die einzige Partei, die für die Schlüsselverteilung in Frage kommt.

3.5.1 Beschreibung der Beziehungen

Im Rahmen dieses Modells müssen die Beziehungen zwischen folgenden Rollen beachtet werden: Endnutzer, Zugangsnetzbetreiber, kontaktierter ISP, RSP, Heim-ISP, Unternehmen und PKI Dienstanbieter. Diese stehen jeweils paarweise in unterschiedlicher Beziehung zueinander. Diese Beziehungen entsprechen im Wesentlichen denen des Seamless Roaming VPN Modells mit internem HA. Die folgende Beziehung kommt noch hinzu:

- Der RSP und das Unternehmen vereinbaren, dass der RSP die Dienste eines HA bereitstellt. Diese Vereinbarung könnte auch den ISP betreffen, wenn es sich um ein kleines Unternehmen handelt, das nicht gleichzeitig die Rolle eines ISPs spielt.
- Das Unternehmen und der ISP unterzeichnen Verträge, Das Unternehmen bezahlt dabei Geld an den ISP für die Nutzung der vom ISP angebotenen Dienste.
- Das Unternehmen und der RSP schließen einen Vertrag dahingehend ab, dass der RSP Zertifikate der Mitarbeiter des Unternehmens akzeptiert, auch wenn sie von einer unternehmenseigenen CA ausgestellt werden.
- Der ISP kooperiert mit dem RSP auf vertraglicher Basis, um seinen Kunden einen Roaming Dienst anzubieten.
- Für die Abrechnungen zwischen den ISPs definiert der RSP Regeln, nach denen die ISPs sich zu richten haben. Der Heim-ISP stellt die Rechnungen an seine Kunden und gibt das Geld zum kontaktierten ISP entweder direkt oder über den RSP weiter.
- Das Unternehmen stellt dem RSP die Beschreibung seiner Sicherheitsstrategie zu Verfügung.
- Die ISPs stellen Beschreibungen ihrer Strategien für einander bereit. Der kontaktierte ISP bietet dabei entsprechend der Strategie des Heim-ISPs Dienste

an. Der RSP kann hier involviert werden, wenn die Strategie des Heim-ISPs bei ihm gespeichert ist oder der RSP Zertifikate oder Attributzertifikate ausstellt, welche die Beschreibungen der Strategien enthalten.
- Der Zugangsnetzbetreiber hat einen Vertrag mit dem kontaktierten ISP.
- Der Zugangsnetzbetreiber stellt seinen Zugangspunkt dem Endnutzer entweder mit oder ohne Vertrag zu Verfügung.
- Der RSP kann mit einem PKI Betreiber zusammenarbeiten, wenn er die entsprechenden Dienste nicht selbst anbietet.
- In dem Fall, dass das Unternehmen seine eigene CA betreibt, akzeptiert der RSP die Zertifikate, welche von der CA ausgestellt wurden

3.5.2 Aktivitäten

Die Hauptaktivitäten, der in das hier zugrunde gelegte Geschäftsmodell involvierten Rollen, werden nun beschrieben. Da die Rolle des Unternehmen in diesem Modell existent ist, kann die CA entweder vom Unternehmen oder vom RSP oder von einem Dritten, der mit einem dieser beiden eng zusammenarbeitet betrieben werden. Unternehmen können eine Unternehmens CA betreiben und noch zusätzliche PKI bezogene Dienste, wie z.B. CRLs anbieten. Dieser Fall ist denkbar, aber nicht sehr wahrscheinlich, wenn man von kleineren Unternehmen ausgeht. Die Aktivitäten sind im Einzelnen:

- Der RSP ist dafür verantwortlich, Zertifikate für Endnutzer auszustellen, der RSP betreibt seine eigene CA oder nutzt Dienste einer dritten Partei.
- Der Endnutzer erhält seine Zertifikate über sein Unternehmen entweder von der Unternehmenseigenen CA oder von der CA des seinem Unternehmen assoziierten RSP
- Der RSP verwaltet bzw. betreibt den HA des Endnutzers.
- Der RSP ist dafür verantwortlich Zertifikate für den Endnutzers zu erstellen.
- Der Endnutzer sollte in der Lage sein, Signaturen von ISPs zu überprüfen, die auf Zertifikaten beruhen, die vom RSP ausgestellt wurden. In dem Fall, dass der Endnutzer nicht in der Lage ist, die Signaturüberprüfung durchzuführen, muss der Endnutzer dem ISP der Korrektheit der signierten Daten vertrauen. Falls kein Vertrauen gegeben ist, kann der Endnutzer den Prozess abbrechen.
- Der RSP generiert die VPN Sitzungsschlüssel für die Kommunikation zwischen den Unternehmen und dem Endnutzer.
- Die Authentifizierung des Endnutzers wird vom kontaktierten ISP durchgeführt. Wenn die Authentifizierung nicht möglich ist, wird sie an den RSP delegiert. Die Delegation der Authentifizierung ist z.B. notwendig, wenn der kontaktierte ISP nicht in der Lage ist, einen bei der Authentifizierung erforderlichen Algorithmus auszuführen oder auf Grund von staatlichen Regulierungen die gegebenen Schlüssel wegen ihrer Länge nicht einsetzen darf.
- Der RSP authentifiziert den Nutzer, wenn er ein von einem Unternehmen ausgestelltes Zertifikat hat, welches dem kontaktierten ISP unbekannt ist. Im Allgemeinen kann man nicht davon ausgehen, dass ein kontaktierter ISP einen Validierungspfad zu einer unternehmenseigenen CA herstellen kann.

- Ein kontaktierter ISP entscheidet prinzipiell über den Zugriff auf seine Dienste unabhängig von der Autorisierungsinformation. Ob ein Endnutzer einen Dienst in Anspruch nehmen darf oder nicht hängt jedoch von der dem ISP gegebenen entsprechenden Autorisierungsinformation ab. Die Autorisierungsinformation wird entweder vom Unternehmen, vom Heim-ISP oder vom RSP gemäß deren Strategie zu Verfügung gestellt.
- Im Falle von auf Zertifikatserweiterungen basierender Autorisierung, wie unten in Kapitel 7 beschrieben, können die Zugangskontrollentscheidungen vom kontaktierten ISP selbst getroffen werden.
- In dem Fall, dass die Autorisierungsinformation nur in Datenbanken gespeichert ist, sendet der kontaktierte ISP eine Anfrage zum RSP bezüglich der Zugangskontrollentscheidungen, welche vom kontaktierten ISP umzusetzen sind.
- Der kontaktierte ISP sammelt Abrechnungsinformation für vom Endnutzer in Anspruch genommene Dienste und leitet diese Information zum Heim-ISP weiter.
- Der Heim-ISP stellt dem Unternehmen passend zu der gesammelten Abrechnungsinformation eine Rechnung und transferiert einen entsprechenden Anteil zum kontaktierten ISP entsprechend dem zugrunde liegenden Vertrag. Das Geld kann hierbei über den RSP fließen oder nicht.

3.5.3 Erlöse

Das in Abschnitt 3.3.3 bereits gesagte gilt auch hier.

3.5.4 Vorteile

Die Vorteile, welche die einzelnen Rollen von dem hier entwickelten Geschäftsmodell haben, werden im Folgenden skizziert.

3.5.4.1 Vorteile für den RSP

Der RSP hat die Möglichkeit, Einkommen für das Anbieten seines Dienstes zu generieren. Dadurch, dass er im Vergleich zum vorigen Roaming VPN Modell mit internem HA auch noch die Funktionalität des HA bereitstellt, kann er darüber hinaus.
Der RSP hat die Möglichkeit, Einkommen für das Anbieten seines Dienstes zu generieren.

3.5.4.2 Vorteil für Unternehmen / Endnutzer

Folgende Vorteile ergeben sich für das Unternehmen und den Endnutzer:

- Die Nutzbarkeit steigt durch die Mobilitätsunterstützung.
- Es besteht keine Notwendigkeit, mehrere Vertragsverhältnisse mit verschiedenen ISPs einzugehen.
- Dem Unternehmen bzw. Nutzer wird nur eine Rechnung gestellt und er muss nur eine Zahlung zu einem Partner durchführen.
- Der Aufbau einer VPN Verbindung zum Unternehmen durch die Mitarbeiter ist möglich und kann sogar erzwungen werden
- Mobilitätsunterstützung während der Kommunikation durch übergangsloses Roaming

- Unterstützung von Flexibilität bezüglich der dem Zugangsnetz zugrunde liegenden Technologie.

3.5.4.3 Vorteil für die ISPs

Folgende Vorteile ergeben sich für dis ISPs:

- Die Anzahl der Kooperationen, in die ISPs involviert sind, lässt sich von n(n-1)/2 auf n im Falle von n ISPs reduzieren.
- Es können mehr Endnutzer erreicht werden ohne mehr Vertragsverhältnisse eingehen zu müssen. Die ist ein Wettbewerbsvorteil in dem Sinne, dass die Attraktivität durch die das „Roaming“ Angebot steigt, wodurch mehr Kunden gewonnen werden können.
- Die zusätzlich zum ISP Roaming Modell gegebene Möglichkeit für Kunden des ISPs ein VPN aufzubauen, erhöht die Attraktivität des ISPs. Damit eröffnet sich eine weitere Einnahmequelle für den ISP.
- Es bietet sich die Möglichkeit eines Wertetransfers zwischen ISPs, die keine Vertragspartner sind, die sich nicht einmal kennen.

3.5.4.4 Vorteil für Zugangsnetzbetreiber

Der Zugangsnetzbetreiber hat die Möglichkeit, Einkommen für die Bereitstellung seines Dienstes zu generieren.

3.6 Vertrauensmodell und Sicherheitsanforderungen

Bei der Entwicklung von Sicherheitslösungen muss man die Vertrauensbeziehungen zwischen den beteiligten Parteien bezüglich ihrer spezifischen Aktivitäten analysieren. In dieser Arbeit wird vertrauen so verstanden, dass es immer zwischen zwei Parteien hinsichtlich einer Aktivität besteht. Wenn die Partei P einer anderen Partei P' hinsichtlich der Aktivität a vertraut, dann ist P abhängig von P' bezüglich der Aktivität a. Das bedeutet, dass durch P keine Überprüfung der korrekten Ausführung der Aktivität a durch P' erfolgt. Wenn Vertrauen in P' bezüglich der Aktivität a gesetzt wird, bedeutet das nicht automatisch, dass gleichzeitig Vertrauen in P' bezüglich einer anderen Aktivität a' besteht.

Wenn kein Vertrauen zwischen den Parteien gegeben ist, dann werden technische Mechanismen benötigt, um ehrliche Nutzer vor negativen Konsequenzen zu schützen; Diese Mechanismen erlauben entweder eine *a posteriori* Überprüfung der Aktivitäten einer Partei, um bösartigen bzw. betrügerischen Parteien das fehlerhafte Verhalten nachweisen zu können oder eine Vorbeugung vor unerwünschtem Verhalten - einen *a priori* Schutz.

Es gibt unterschiedliche Stufen der Überprüfung, welche mit technischen Mitteln erreicht werden können. Jeweils hinsichtlich Zeit, Umfang, Prüfer und Intensität können wie folgt Stufen unterschieden werden:

- Bezüglich des Faktors Zeit kann man sofortige und „verzögerte“ Überprüfung unterscheiden. Wen es um die Frage der Autorisierung geht, ist in der Regel eine

sofortige Überprüfung erforderlich. Wenn Abrechnungsdaten jedoch innerhalb eines bestimmten Zeitintervalls überprüft werden können – z. B. wegen Reklamationsmöglichkeiten – kann eine Überprüfung zeitlich verzögert stattfinden.

- Bezüglich des Umfangs kann man zwischen einer vollständigen Überprüfung und einer stichprobenartigen Überprüfung unterscheiden. Im Falle der vollständigen Überprüfung prüft die überprüfende Instanz in jedem einzelnen Fall die Korrektheit einer Anfrage. Im Falle der stichprobenartigen Überprüfung entscheidet eine Partei nach dem Zufallsprinzip, ob eine Überprüfung stattfinden soll oder nicht.
- Bezüglich der Prüfer kann man zwischen Überprüfung durch die involvierten Parteien und Überprüfung durch eine dritte Partei unterscheiden. Beispiele sind die selbst durchgeführte Authentifizierung und die delegierte Authentifizierung.
- Bezüglich der Intensität der Überprüfung kann man zwischen starken Beweisen und schwachen Beweisen unterscheiden. Authentifizierungsprotokolle, die Daten mit aktuellst möglichen Werten verwenden sind stärker, als solche ohne derartige Werte bzw. mit älteren Werten.

Der wesentliche Punkt ist die Authentifizierung. Ein zusätzlicher Schritt und sehr wichtige Designanforderung ist, dass die für die Nutzerauthentifikation benötigte Zeit kurz sein sollte. Das bedeutet, dass die Anzahl der Nachrichten, welche ausgetauscht werden müssen, möglichst niedrig sein muss. Außerdem sollten die gesamten Prozesskosten möglichst gering sein.

Generell existiert das Sicherheitsproblem, dass miteinander kommunizierende Parteien falsche Identitäten annehmen können. Daher kann davon ausgegangen werden, dass eine angesprochene Partei nicht einfach auf die Richtigkeit einer vom Kommunikationspartner erklärten Identität vertraut. Dementsprechend ist das Ziel, hier im Folgenden Protokolle dafür zu entwickeln und dabei den entsprechenden Nachrichtenaustausch aufzuzeigen. Bei diesem Ziel wird im Folgenden vorausgesetzt, dass die ISPs und die RSPs in dem hier vorgestellten Modell in der Lage sind, sich gegenseitig zu authentifizieren. Des Weiteren etablieren diese Parteien sichere Kommunikationskanäle in dem Sinne, dass die Veränderungen von ausgetauschten Nachrichten bemerkt werden können. Weiterhin muss die gesamte Kommunikation vertraulich sein, so dass Parteien, welche Zugang zum Netz haben, vorsorglich keine Möglichkeit haben sollten, Inhalte der ausgetauschten Nachrichten zu lesen.

Wenn zusätzlich eine sichere Verbindung, beispielsweise zu einem Unternehmen aufgebaut werden soll, stellt sich die zusätzliche Anforderung, dass das Unternehmen seinen Angestellten - den Endverbraucher - als solchen identifizieren muss. Des Weiteren wird beim Aufbau von VPNs immer eine beidseitige Authentifizierung zwischen Unternehmen und RSP benötigt. Dies ist unbedingt erforderlich, da die zur Authentifizierung benötigte Information an das Unternehmen weitergeleitet wird. Weiterhin müssen Schlüssel ausgetauscht werden, sowohl zwischen Unternehmen und

RSP als auch zwischen Endkunden und RSP. Die Kommunikation zwischen diesen Parteien muss also auch vertraulich sein. Da der RSP die VPN Schlüssel für das Unternehmen und die umherreisenden Mitarbeiter desselben erzeugt, müssen beide dem RSP dahingehen vertrauen, dass er die Schlüssel nicht missbraucht. Des Weiteren müssen Endverbraucher und Unternehmen dem RSP dahingehend vertrauen, dass er die zwischen ihnen ausgetauschten Informationen nicht missbraucht oder verändert.

Die gesamte Kommunikation zwischen Unternehmen und „umherreisenden" Angestellten bzw. Mitarbeitern ist grundsätzlich als vertraulich zu betrachten und sollte daher wie bei VPN Verbindungen üblich verschlüsselt werden.

Zwischen ISP/RSP und Zugangsnetzbetreiber auf der einen Seite und Endverbraucher auf der anderen Seite gibt es keine Vertrauensbeziehung bezüglich der erklärten Identitäten. Daher ist ein entsprechender Authentifizierungsmechanismus nötig, der es diesen Parteien erlaubt, die Identität des Endverbrauchers bzw. Endnutzers zu überprüfen. Aufgrund der Geschwindigkeitsanforderung an die Authentifizierung, sollte der Nutzer von allen Parteien authentifiziert werden können selbst aber nur ein Authentifizierungsprotokoll abwickeln müssen. Da der Endnutzer nicht sicher sein kann mit welcher Partei er gerade kommuniziert, ist es für den Endnutzer erforderlich, dass der ISP und der Zugangsnetzbetreiber sich ihm gegenüber ebenfalls authentifizieren.

Die öffentliche CA bzw. der RSP vertrauen dem ISP, mit welchem der Endnutzer einen Vertrag hat, dahingehend, dass der ISP die Daten, welche die Zertifikate beinhalten, korrekt aufgenommen und verarbeitet hat. Der ISP füllt die Rolle der „Registration Authority" RA aus. Eine CA ist unbedingt erforderlich, da die in Kapitel 4 beschriebene Authentifizierung auf Zertifikaten beruht. Wenn man mehrere Milliarden Kunden von einer Vielzahl Mobilfunkanbietern, Internet-Service-Providern und Mitarbeiter tausender Unternehmen denkt, die in einer Struktur zusammengeführt bzw. berücksichtigt werden müssen, dann eignet sich das „Web of trust" im Vergleich zu einer hierarchischen PKI Struktur auf Grund der in Kapitel 2.5 aufgeführten Nachteile nicht. Der Aufbau eines weltweiten Vertrauensnetzes, wie es für das Roaming erforderlich ist, wäre auch vergleichsweise umständlich für einzelne Nutzer. Beim Roaming muss eine klare Entscheidung darüber, ob ein neues Zugangsnetz vertrauenswürdig genug dahingehend ist, dass der Nutzer sein Endgerät mit ihm verbindet, ohne Nutzerinteraktion schnell getroffen werden. Eine Abstufung von Vertrauen, wie sie beim „Web of trust" vorgesehen ist, ist daher, wenn es um Roaming zu einem neuen Zugangsnetz geht, nicht sinnvoll.

Weiterhin wird vorausgesetzt, dass alle Parteien der CA dahingehend vertrauen, dass sie Zertifikate korrekt ausstellt, dass sie private Schlüssel sicher und CRLs korrekt verwaltet. Zur Vereinfachung wird im Folgenden von einer einzigen CA ausgegangen. Alle Parteien, welche mit einem gegebenen RSP in Beziehung stehen, sind in der Lage ein Zertifikat der öffentlichen CA zu erkennen.

Der Heim-ISP vertraut dem kontaktierten ISP nicht bezüglich der Authentifizierung des Endnutzers. Die beiden ISPs könnten Konkurrenten sein oder einander sogar völlig unbekannt sein, so dass keine Basis für eine Vertrauensbeziehung welcher Art auch immer besteht. Der Heim-ISP wird daher vermutlich seine Endkunden selbst authentifizieren wollen. Dies darf nicht im Gegensatz zur Hochgeschwindigkeitsanforderung stehen. Des Weiteren, stellt der Heim-ISP Zugriff auf Daten für seine Kunden zu Verfügung, wie z.B. Mailbox Inhalte oder andere persönliche Daten, welche geschützt werden müssen. Für solche Zwecke jedoch ist das Ergebnis der Authentifizierung eine wichtige Information für die Zugangskontrollentscheidung des kontaktierten ISP. Der Heim-ISP könnte Interesse daran haben entweder den Endnutzer selbst zu authentifizieren oder zumindest das Ergebnis der initialen Authentifizierung zu überprüfen.
In dieser Arbeit wird vorausgesetzt, dass es keine Vertrauensbeziehung zwischen dem kontaktierten ISP und dem Betreiber des Zugangsnetzes hinsichtlich der Authentifizierung gibt. Soweit der Betreiber des Zugangsnetzes keine vertragliche Beziehung mit dem Endnutzer hat, besteht für den Zugangsnetzbetreiber nicht die Notwendigkeit die erklärte Identität des Endnutzers zu überprüfen auch wenn er ihm nicht vertraut. Dies ist schematisch in Abbildung 34 dargestellt.

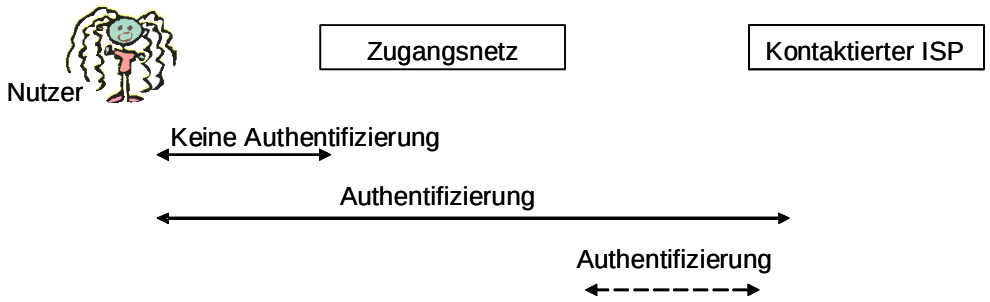


Abbildung 34: Keine Authentifizierung durch den Zugangsnetzanbieter

In einem solchen Fall wird kein Authentifizierungsprotokoll benötigt. Wenn eine Vertragsbeziehung zwischen Zugangsnetzbetreiber und Endnutzer besteht, hat der Zugangsnetzbetreiber ein Interesse daran die erklärte Identität des Endnutzers zu überprüfen. Dabei kann man zwei Fälle unterscheiden. Der kontaktierte ISP kann den Endnutzer selbst authentifizieren, wie in Abbildung 35 dargestellt oder alternativ vom Zugangsnetzbetreiber mit der benötigten Information versorg werden.

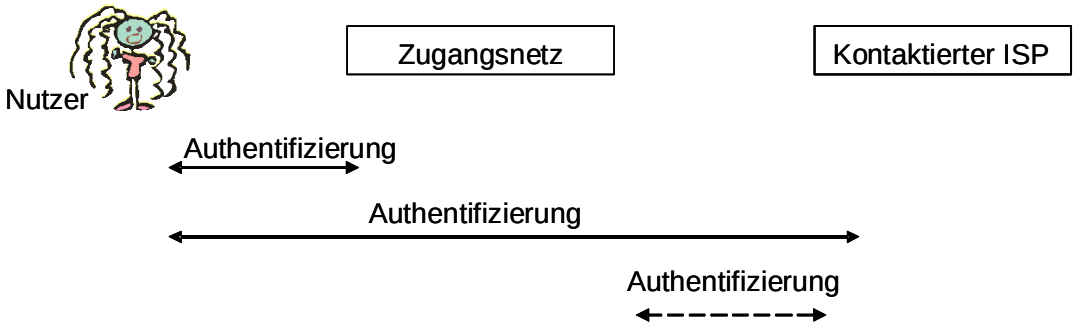


Abbildung 35: Getrennte Authentifizierung durch Zugangsnetzbetreiber und kontaktierten ISP

In dem in Abbildung 35 dargestellten Fall der getrennten Authentifizierung kann die Anforderung, dass der Endnutzer nur einmal in einen Authentifizierungsvorgang verwickelt sein soll, nicht erfüllt werden. Die Abbildung 36 zeigt wie alternativ der kontaktierte ISP vom Zugangsnetzbetreiber mit dem Ergebnis der Authentifizierung versorgt wird.

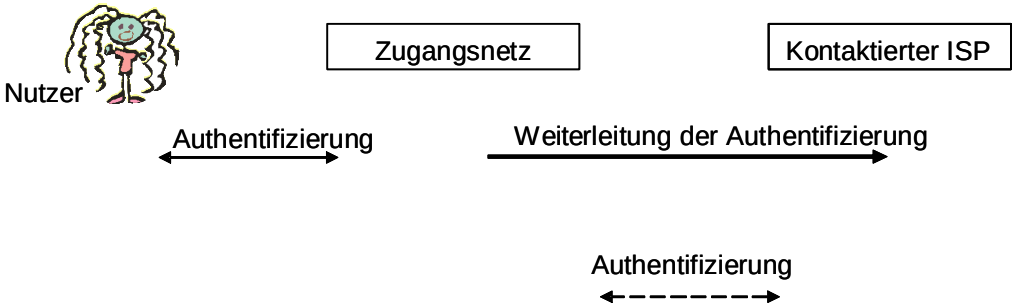


Abbildung 36: Authentifizierung durch Zugangsnetzbetreiber und Weiterleitung des Authentifikationsergebnisses zum kontaktierten ISP

Der in Abbildung 36 gezeigte Fall bietet sich vor allem dann an, wenn die Rollen des Zugangsnetzbetreiber und des ISPs von demselben Unternehmen ausgefüllt werden. Das hat den Vorteil, dass der Endnutzer nur einmal mit einer der beiden Rollen ein Authentifizierungsprotokoll abwickeln muss.

In einigen Fällen mag das Unternehmen eine eigene PKI betreiben. Das umfasst RA, CA und die Bereitstellung von Sperrinformationen - CRLs. Dabei stellt sich die Frage, ob Zertifikate, die von der unternehmenseigenen CA ausgestellt wurden, von anderen Parteien erkannt werden. Es wird hier vorausgesetzt, dass zumindest der RSP und das Unternehmen, welches die CA betreibt, die Zertifikate erkennen können.

Bei den ISPs und den Betreibern der Zugangsnetze sieht die Situation anders aus bezüglich der Überprüfung der Zertifikate. Es kann sein, dass sie nicht in der Lage sind, für die von jedem Unternehmen ausgestellten Zertifikate einen Pfad zu einem Vertrauensanker zu konstruieren.

Um übergangsloses bzw. unterbrechungsfreies (“seamless“) Roaming, basierend auf mobile IP, anzubieten, muss man sich die Frage stellen, wo man den HA des Endkunden unter Berücksichtigung des passenden Sicherheitslevels platziert. In diesem Zusammenhang beschäftigen wir uns mit den Möglichkeiten der Position des HAs. Der HA kann entweder beim RSP oder beim Unternehmen angesiedelt sein. Bei beiden Möglichkeiten werden hier entsprechende Vertrauensbeziehungen vorausgesetzt. Das bedeutet, dass in unserem Szenario der Endnutzer dem RSP oder Unternehmen vertraut seinen HA korrekt zu verwalten bzw. zu administrieren. In dem Fall dass der HA sich beim RSP befindet, muss das Unternehmen dem RSP ebenfalls diesbezüglich vertrauen schenken.

In dieser Arbeit werden vorwiegend die die Authentifizierung betreffenden Sicherheitsaspekte bzw. Sicherheitsanforderungen betrachtet. Aus den Geschäftsmodellen resultierenden wie oben im Text dargestellt hierzu die folgenden Anforderungen:

- Der Endnutzer soll nicht nur authentifiziert werden können, sondern seinen Kommunikationspartner auch selbst authentifizieren können. Es sollte also eine beidseitige Authentifizierung gegeben sein.
- Der Authentifizierungsvorgang soll möglichst wenig Zeit in Anspruch nehmen (Hochgeschwindigkeitsanforderung)
- Möglichkeit des Heim-ISP seine Kunden selbst zu authentifizieren ohne dem kontaktierten ISP vertrauen zu müssen, soll gegeben sein.
- Möglichkeit zur vertraulichen Kommunikation zwischen Angestelltem und Unternehmen sollte gegeben sein.
- Möglichkeit eine Unternehmenseigene PKI einzubinden sollte gegeben sein.
- Bei Aufbau einer sicheren Verbindung zwischen dem Endnutzer als Angestelltem und seinem Unternehmen müssen der Angestellte und das Unternehmen sicher sein, mit der richtigen Partei zu kommunizieren.
- RSP muss Nutzer und Unternehmen authentifizieren können.

Aspekte, welche die Autorisierung und Abrechnung betreffen, werden auch berücksichtigt, wie an den jeweiligen Stellen beschrieben.

Nachdem in diesem Kapitel die Vertrauensbeziehungen und Anforderungen ermittelt und dargestellt sind, wird im folgenden Kapitel eine Architektur entwickelt, welche den aus den Geschäftsmodellen resultierenden Anforderungen gerecht wird.

4 Architektur

Das Ziel dieser Arbeit ist es, eine Sicherheitsarchitektur für die beschriebenen „RSP“-Geschäftsmodelle zu entwickeln.

Die in Kapitel 2.4.3 gezeigte Interworking Architektur von 3GPP wird den in Kapitel 3.1 beschriebenen Rollen der Geschäftsmodelle nicht gerecht, da dort nur WLAN und UMTS zusammenspielen und die Rollen der ISPs und RSPs nicht betrachtet sind. Dies ist jedoch für ein generisches Roaming Modell unbedingt erforderlich. Die folgende Abbildung 37 zeigt das hierzu entwickelte Modell:

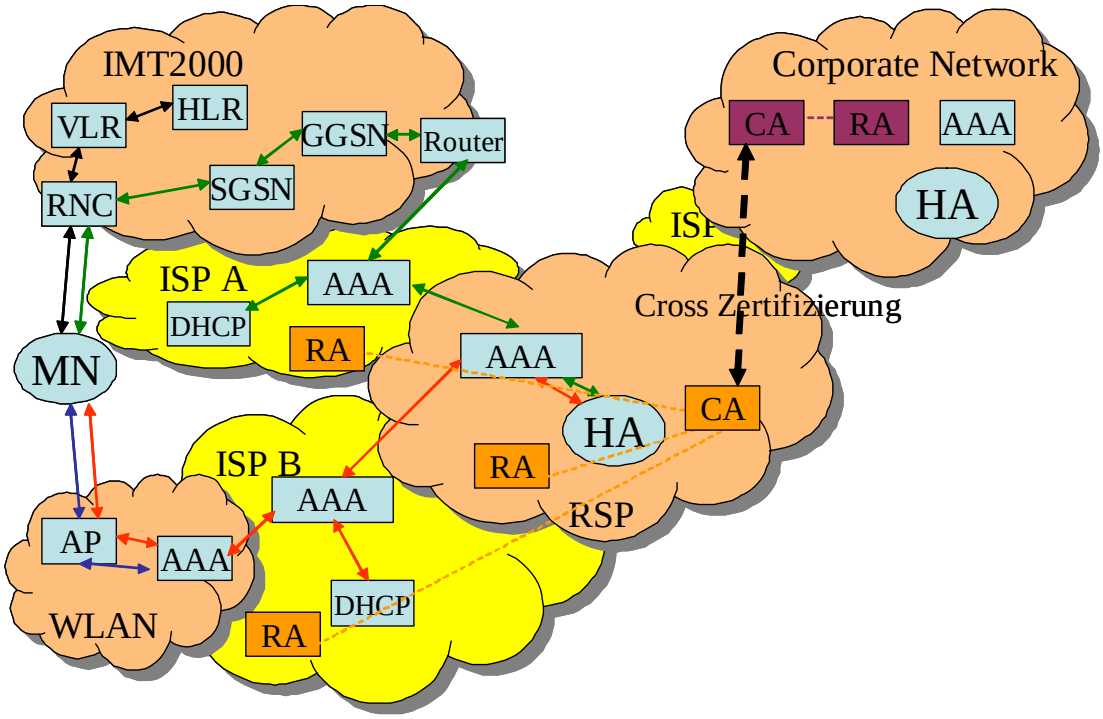


Abbildung 37: Architektur für Roaming mit WLAN und UMTS als Zugangsnetzen

Die farbigen Pfeile in der Abbildung 37 zeigen den während des Vorgangs der Authentifizierung verlaufenden Nachrichtenfluss. Die schwarzen Pfeile zeigen den Nachrichtenfluss bei der Authentifizierung innerhalb von Mobilfunknetzen. Die blauen Pfeile zeigen den Nachrichtenfluss bei WLANs. Stand der Technik ist hier eine Authentifizierung, die im Rahmen des EAP-TLS Protokolls unter Einsatz eines Radius oder Diameter [CAG+02, CLGZ02, CZP+01, HZ02, Int02] entwickelt wird. Diese beiden Authentifizierungsvorgänge werden durchgeführt, wenn im Zugangsnetz zusätzlich zum ISP eine getrennte Authentifizierung durchgeführt wird, wie in Abbildung 35 dargestellt, oder das Zugangsnetz den Endnutzer authentifiziert und das Ergebnis weiterleitet, wie in

Abbildung 36 gezeigt. Die Authentifizierung des Endnutzers kann auch, wie in Abbildung 34 dargestellt, ausschließlich durch den ISP erfolgen. Die grünen Pfeile zeigen den Nachrichtenfluss bei der Authentifizierung mit einem IMT2000 konformen Zugangsnetz, wie z.B. einem UMTS Netz und die roten Pfeile den Fluss über ein WLAN Zugangsnetz.

Zum besseren Verständnis sind im Folgenden einzelne Teile des Modells dargestellt und beschrieben. Abbildung 38 zeigt den Nachrichtenfluss bei der Authentifizierung in einem IMT2000 Netz.

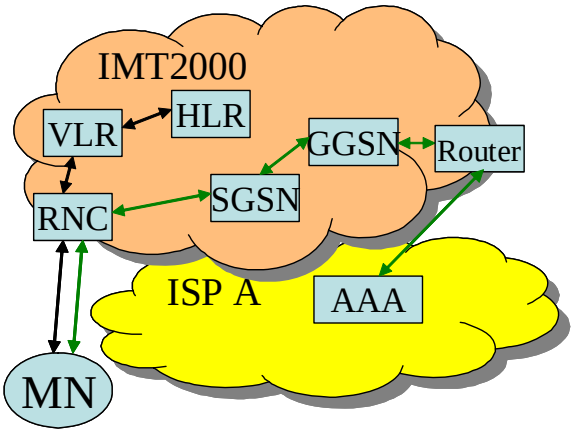


Abbildung 38: Nachrichtenfluss bei der Authentifizierung in einem IMT2000 Netz

Das „Home Location Register“ (HLR) ist im Wesentlichen eine Datenbank des Betreibers zur Verwaltung von Teilnehmerinformationen. Es enthält üblicherweise:

- Daten über den Teilnehmer wie z. B. Art und Umfang abonnierter Dienste, max. zulässige Übertragungsrate, u. s. w..
- Es enthält die Identifikation des Endgeräts bzw. die International Mobile Equipment Identity (IMEI),
- die Rufnummer des Teilnehmers bzw. die Mobile Station International ISDN Number (MSISDN) und
- die Visitor Location Register (VLR)-Nummer, unter der der Teilnehmer registriert ist für die schnelle Erreichbarkeit des Teilnehmers.

Das VLR ist ebenfalls eine Datenbank des Betreibers, die mit einem oder mehreren Mobile Services Switching Center (MSCs) verbunden ist. Die MSCs dienen lediglich zum Verteilen der Signale. Seine Funktion beinhaltet:

- Steuerung des Verbindungsaufbau zum Endgerät
- Registriert jeden Teilnehmer innerhalb einer bestimmten geog. Zone, Location Area (LA)
- Beim LA Wechsel wird der VLR neue LA mitgeteilt
- Speicherung TMSI des Endgeräts. Die „Temporary Mobile Subscriber Identity“ (TMSI) dient zur Bekämpfung betrügerischen Abfangens und Verwendens der IMSI.

- Speichert aktuellen Standort - den LA - des Teilnehmers

Der Radio Network Controller (RNC) steuert mehrere so genannte „Node-B“s, die als Basis-Stationen über Ihrer Antennen die Funkkommunikation zum Mobile-Node (MN) durchführen.

Node-B und MSC sind in der obigen Abbildung nicht dargestellt, da sie im Wesentlichen nur Funksignale erzeugen, senden und empfangen und damit bei Betrachtung des Authentifizierungsvorganges von untergeordneter Bedeutung sind. Der paketorientierte Datenverkehr wird von dem Serving GPRS Support Node (SGSN) abgewickelt. Das Routing ins Internet übernimmt der Gateway GPRS Support Node (GGSN).

Die beiden letzten in Abbildung 38 dargestellten Komponenten sind ein Router an der Grenze des IMT2000 Netzes und der Authentifizierungs-, Autorisierungs- und Abrechnungs- (AAA) Server des ISPs, zu dem der Router die entsprechenden Pakete leitet.

Die schwarzen Pfeile stellen den Fluss bei der Authentifizierung innerhalb des IMT2000 Netzes dar. Der genaue Authentifizierungsvorgang ist in Abschnitt 2.2.9 für UMTS - einer IMT2000 Ausprägung - bereits beschrieben. Die folgendeAbbildung 39 gibt einen Überblick über diesen Vorgang.

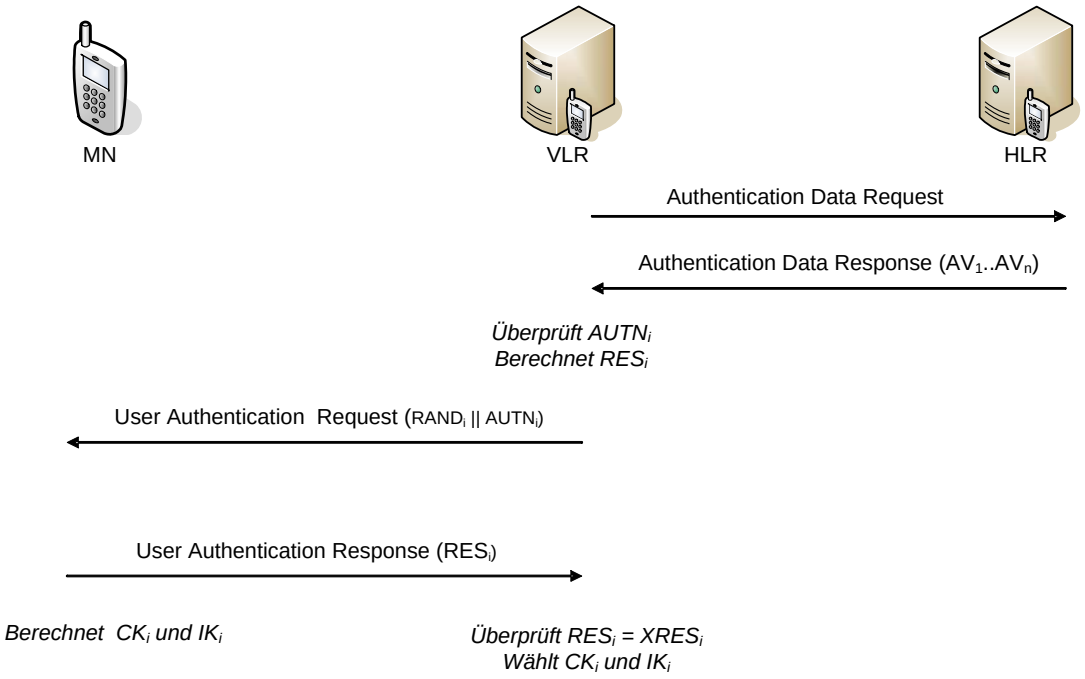


Abbildung 39: 3GPP Authentication and Key Agreement Protocol (AKA)

Die grünen Pfeile zeigen den Nachrichtenfluss bei der Authentifizierung gegenüber einem AAA Server, der in der hier dargestellten Architektur bei einem ISP lokalisiert ist. Für diese Authentifizierung kann prinzipiell jedes geeignete Protokoll eingesetzt werden, welches auch im Internet eingesetzt wird. Die folgende Abbildung 40 zeigt das Prinzip des Internetzugangs über UMTS oder auch GSM. Der Protokoll-Stack der relevanten Komponenten über die das Authentifizierungsprotokoll läuft, wird hierbei dargestellt.

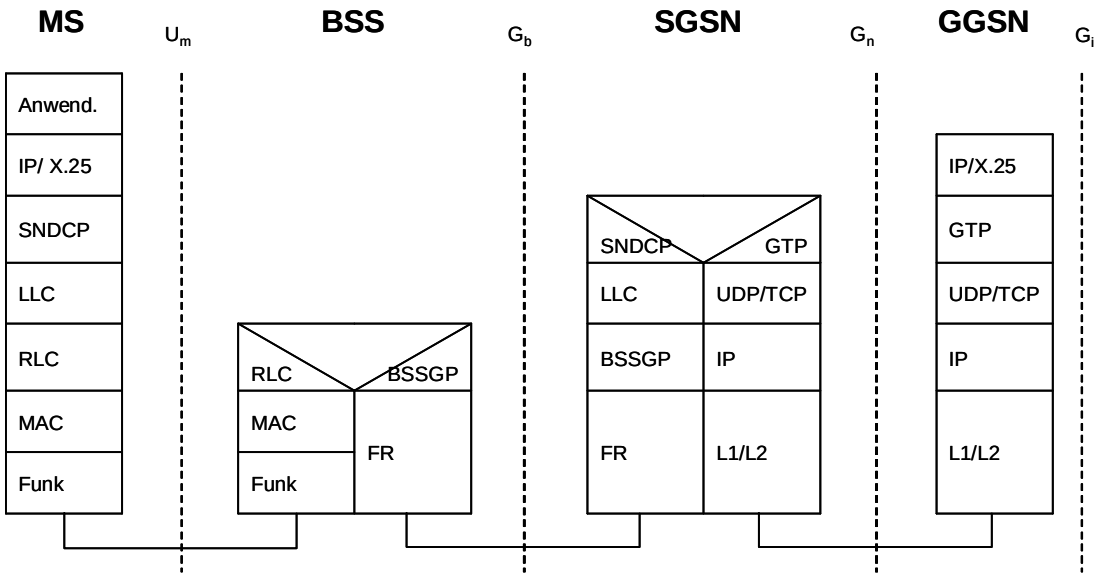


Abbildung 40: GPRS Referenzmodell

Details hierzu finden sich in [ETSI98]. Zum gegenwärtigen Zeitpunkt ist zunächst eine Authentifizierung innerhalb des Zugangsnetzes, wie oben beschrieben über AKA notwendig, um danach eine zweite Authentifizierung zwischen Nutzer und ISP durchführen zu können.

Die folgende Abbildung 41 zeigt die möglichen Informationsflüsse beim Authentifizie-

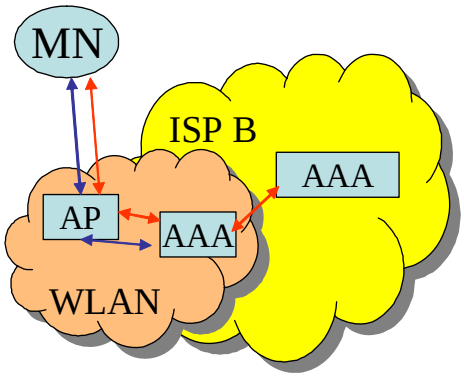


Abbildung 41: Nachrichtenfluss bei der Authentifizierung über ein WLAN

rungsvorgang über ein WLAN Zugangsnetz. Bei einem WiMAX Zugangsnetz könnte dies genauso aussehen.

Die blauen Pfeile zeigen den Nachrichtenfluss, der vorliegt, wenn der MN vom WLAN selbst authentifiziert wird. Das im Moment am meisten verwendete Protokoll ist hierbei EAP-TLS[23] over Radius. Der Nachrichtenaustausch bzw. der Ablauf der Authentifizierung in diesem Fall wird in der folgenden Abbildung 42 dargestellt.

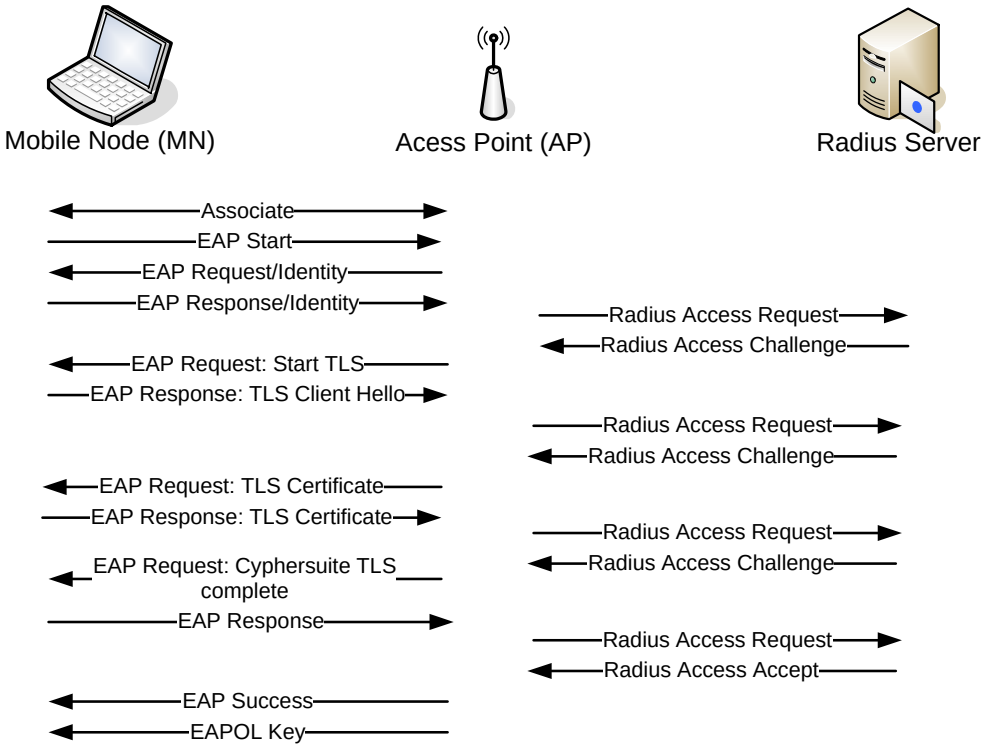


Abbildung 42: EAP-TLS over Radius

Er lässt sich in acht Schritten beschreiben:

1. Schritt: Der MN als Client schickt eine „EAP Start“ Nachricht an einen AP, mit dem er sich assoziiert hat. Der AP beantwortet dies mit einer „EAP Request/Identity“ Nachricht. Der MN antwortet darauf normalerweise mit dem Namen des angemeldeten Nutzers in einer „EAP Response/Identity“ Nachricht. Falls kein Nutzer angemeldet ist, wird der Name des Rechners bzw. des MN übertragen. Dies wird in einer Radius „Access-Request“ Nachricht vom AP an den Radius Server weitergeleitet.
2. Schritt: Der Radius Server schickt nun eine „Radius Access-Challenge“ Nachricht an den AP, welche TLS als EAP-Typ enthält. Dies wird vom AP an den MN weitergeleitet. Der MN schickt eine „EAP Response“ Nachricht an den AP,

[23] Extensible Authentication Protocol - Transport Layer Security

welche bei auf TLS gesetztem EAP-Typ eine Client Hello Nachricht eines TLS Handshakes enthält. Der AP leitet diese Nachricht in Form einer „Radius Access Request“ Nachricht an den Radius Server weiter

3. Schritt: Der Radius Server schickt nun seine Zertifikatskette an den AP, welcher diese an den MN weiterleitet
4. Schritt: Der MN schickt seinerseits seine Zertifikatskette an den AP, welcher diese an den Radius Server weiterleitet.
5. Schritt: Der Radius Server sendet eine „EAP-Request“ Nachricht mit der Ciphersuite und der Aussage, dass der TLS-Nachrichtenaustausch komplett ist. Der AP leitet die Nachricht zum MN weiter.
6. Schritt: Der MN antwortet mit einer „EAP Response“ Nachricht, welche vom AP an den Radius Server weitergeleitet wird und beendet damit den TLS Handshake.
7. Schritt: Der Radius Server schickt nun eine „Radius Access Accept“ Nachricht, welche eine „EAP-Success“ Nachricht und zwei Schlüssel enthält an den AP. Die beiden Schlüssel werden aus im Rahmen des TLS Handshake generierten Daten erzeugt. Der AP verwendet den einen Schlüssel – den „unicast key“ - zum verschlüsseln und den anderen Schlüssel zum signieren bei der Kommunikation mit dem MN. Die „EAP-Success“ Nachricht schickt er ohne die Schlüssel an den MN weiter.
8. Schritt: Der AP erzeugt zum Schluss eine „EAPOL Key“ Nachricht, welche einen aus einer Zufallszahl oder einem vorgegebenen Wert erzeugten globalen Schlüssel enthält, und schickt diese mit dem vom Radius Server erhaltenen „unicast key“ verschlüsselt an den MN. Der MN hat den „unicast key“ ebenfalls aus den beim TLS Handshake ausgetauschten Informationen erzeugt und ist so in der Lage den globalen Schlüssel aus der „EAPOL Key“ Nachricht zu extrahieren. Mit diesem globalen Schlüssel wird a sofort die Kommunikation auf der Luftschnittstelle zwischen AP und MN verschlüsselt. Jetzt kann der MN eine IP Adresse z. B. über DHCP erhalten.

Die roten Pfeile zeigen den Nachrichtenfluss bei der Authentifizierung durch einen ISP. Dies entspricht dem gerade beschriebenen Vorgang mit dem Unterschied, dass sich der zur Authentifizierung verwendete Radius Server beim ISP befindet, der Nachrichten über einen als Proxy arbeitenden Radius Server im WLAN austauscht. Der Authentifizierungsvorgang ist dabei prinzipiell gleich. Zwischen dem AP und dem authentifzierenden Radius Server befindet sich nur ein zusätzlicher Proxy, der die Nachrichten vom AP zum Radius Server und umgekehrt weiterleitet, wie in der folgenden Abbildung 43 dargestellt.

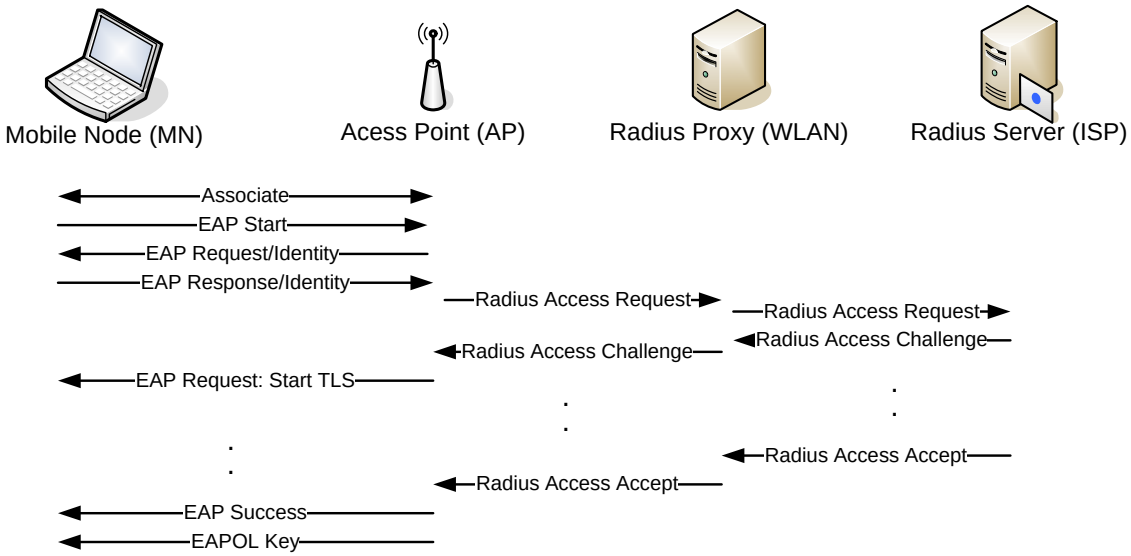


Abbildung 43: EAP over Radius mit Proxy

Die Abbildung 37 enthält darüber hinaus noch das Netz des RSP und ein Unternehmens-Netz beide verfügen auch über einen AAA Server und beide können Home Agents (HA), wie in Kapitel 2.3.1 beschrieben, verwalten. Die genaue Funktionsweise der bei einer PKI Infrastruktur benötigten Komponenten, wie der Certification Authority (CA) und der Registration Authority (RA) wird in Abschnitt 6 beschrieben. Der AAA Server des RSP kann als Radius Server ebenso wie der Radius Server eines ISPs zur Authentifizierung herangezogen werden. Das entspricht dann Abbildung 43 mit einem weiteren Radius Proxy. Der Radius Server des RSP authentifiziert den MN und auch der Radius Server des ISP arbeitet als Proxy.

Im Vergleich zu Abbildung 37 zeigt die folgende Abbildung 44 das 3GPP Modell. Die Rolle des RSP fällt hierbei weg. Der RSP ist zum besseren Vergleich trotzdem in der Abbildung enthalten – allerdings rot durchgestrichen. Weiterhin ist zu erkennen, dass das Corporate Network nicht mehr eingebunden ist.

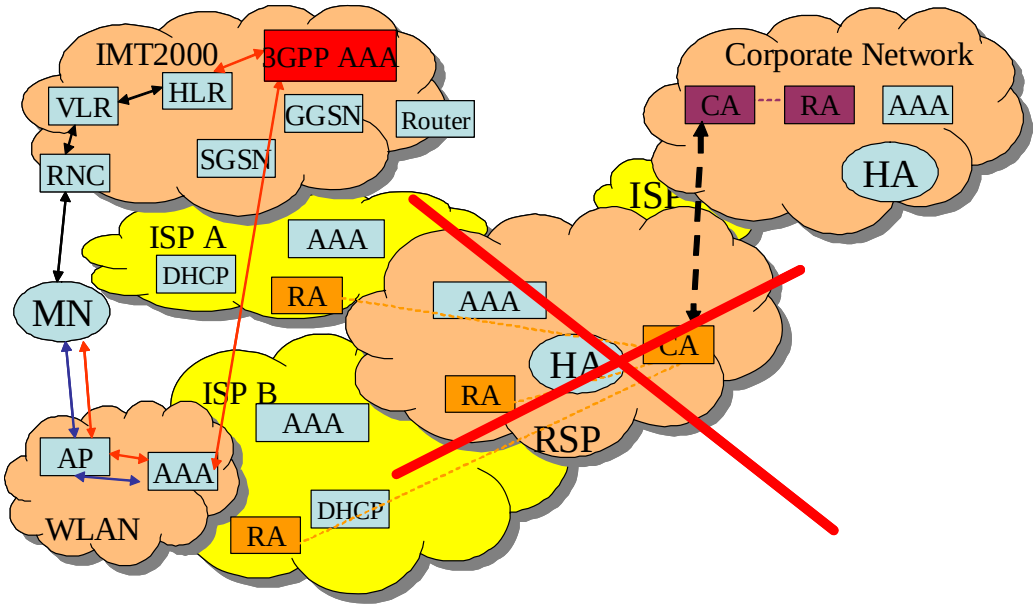


Abbildung 44: 3GPP Interworking Modell

In Abbildung 37 sind nur ein WLAN und ein IMT2000 Netz beispielhaft als Zugangsnetze dargestellt. Prinzipiell können die Zugangsnetze auch auf den anderen in Kapitel 2 beschriebenen Funktechnologien basieren.

Im Weiteren wird zunächst

- In Kapitel 5 eine generische Lösung für die Authentifizierung entwickelt.

Danach werden

- in Kapitel 6 die PKI basierte Authentifizierungslösung,
- in Kapitel 7 die Lösung zur Autorisierung und
- in Kapitel 8 die Lösung für das Single Sign On (SSO)

in Anlehnung an [FLL03] entwickelt.

5 Generische Authentifikationslösung

Zunächst wird im Folgenden eine generische Authentifikationslösung erstellt, die eine sichere Authentifikation bei minimalem Vertrauen der involvierten Parteien gewährleisten soll. Die generische Lösung dient auch dem tieferen Verständnis möglicher Probleme und Anforderungen an eine Sicherheitsarchitektur, welches dazu hilft, später eine auf Standards basierende sichere Lösung für das Roaming zu erhalten.

In diesem Kapitel beschriebene Authentifikationslösung soll den folgenden Anforderungen genügen:

- Die Lösung ist so allgemein wie möglich zu halten, um dem möglichen Einsatz unterschiedlicher Kommunikationstechnologien gerecht zu werden.
- Existierende Lösungen sollten, wenn Möglich eingesetzt werden und idealer Weise für die unterschiedliche Modelle anwendbar sein.
- Standards sollen so weit wie möglich eingesetzt werden.
- Die Entwicklungskosten sollen so niedrig wie möglich sein.

Das zugrunde liegende Vertrauensmodell ist in Abschnitt 3.6 beschrieben. Die Lösung berücksichtigt alle in Kapitel 3 beschriebenen Modelle:

- ISP Roaming Modell
- Roaming mit VPN Zugang Modell
- Seamless Roaming VPN Modell mit internem HA
- Seamless Roaming VPN Modell mit externem HA

Es wird zunächst mit dem einfachsten Modell für ISP Roaming begonnen.

5.1 ISP Roaming Modell

Im ISP Roaming Szenario wird ein Nutzer U („user“), der einen ISP B kontaktiert, um Zugang zum Internet zu erhalten, betrachtet. B authentifiziert dabei U. Es gibt unterschiedliche Gründe, warum U authentifiziert werden soll. Zunächst will B sicher sein, dass er einen entsprechenden Geldwert von U’s Heim-ISP A erhält. A benötigt das Authentifikationsergebnis, wenn Zugang zu speziellen Ressourcen, wie z.B. U’s Mailbox, angefragt wird. Dies bedeutet, dass das Authentifikationsergebnis permanent überprüfbar sein muss, wenn nicht von A dann zumindest vom RSP R, so dass kein weiteres Authentifikationsprotokoll mit U abgewickelt werden muss. Des Weiteren muss sich B gegenüber U authentifizieren. Die Authentifizierung des Nutzers hängt von der Art und Weise ab, auf die der Zugangsnetzbetreiber involviert ist. Hierbei lassen sich die folgenden Fälle unterscheiden (Es werden in Klammern die zugehörigen Kapitelnummern angegeben):

1. Keine Authentifikation durch den Betreiber des Zugangsnetzes
    a. Authentifikation ausschließlich durch den kontaktierten ISP
        i. Delegation der Authentifikation zum RSP
    b. Authentifikation durch den kontaktierten ISP und den Heim-ISP
        i. Delegation der Authentifikation zum RSP

   c. Authentifikation durch den kontaktierten ISP und Weiterleitung der Authentifikationsinformation
      i. Delegation der Authentifikation zum RSP
2. Getrennte Authentifikation durch den Betreiber des Zugangsnetzes und den kontaktierten ISP
   a. Keine zusätzliche Authentifikation durch den Heim-ISP
      i. Delegation der Authentifikation vom kontaktierten ISP zum RSP
   b. Zusätzliche Authentifikation durch den Heim-ISP
      i. Delegation der Authentifikation vom kontaktierten ISP zum RSP
   c. Authentifikation durch den kontaktierten ISP und Weiterleitung der Authentifikationsinformation
      i. Delegation der Authentifikation zum RSP
3. Authentifikation durch den Betreiber des Zugangsnetzes und Weiterleitung des Authentifikationsergebnisses zum kontaktierten ISP
   a. Keine zusätzliche Authentifikation durch den Heim-ISP
      i. Delegation der Authentifikation vom kontaktierten ISP zum RSP
   b. Zusätzliche Authentifikation durch den Heim-ISP
      i. Delegation der Authentifikation vom kontaktierten ISP zum RSP
   c. Authentifikation durch den Betreiber des Zugangsnetzes und Weiterleitung der Authentifikationsinformation vom kontaktierten ISP zum Heim-ISP
      i. Delegation der Authentifikation vom kontaktierten ISP zum RSP

Diese Fälle werden nun im Folgenden näher erläutert.

5.1.1 Keine Authentifikation durch den Betreiber des Zugangsnetzes

5.1.1.1 Authentifikation ausschließlich durch den kontaktierten ISP

Ein umherwandernder Nutzer will die Dienste eines ihm unbekannten ISPs nutzen Er kontaktiert diesen über ein Zugangsnetz, wie z.B. ein WLAN in einem Café, welches von einer dritten Partei - dem Besitzer des Cafés - bereitgestellt wird, um die Attraktivität seines Etablissements zu erhöhen. Der Bereitsteller des Zugangsnetzes ist nicht an der Identität des Endnutzers interessiert, so dass keine Notwendigkeit besteht diesen zu authentifizieren. Die Geldflüsse B und C aus dem vorigen Kapitel entsprechen typischerweise einem solchen Szenario. Zumindest der kontaktierte ISP wird allerdings ein Interesse daran haben den Endnutzer zu authentifizieren, da er zum einen prüfen muss, ob der Nutzer autorisiert ist, die angeforderten Ressourcen zu nutzen, und zum anderen in der Lage sein muss, die Nutzung seiner Ressourcen eindeutig zuzuordnen, um hinterher eine entsprechende Rechnung stellen zu können. In der Abbildung 45 sind die Authentifikationsbeziehungen der involvierten Rollen in dem hier zugrunde gelegten Szenario schematisch dargestellt.

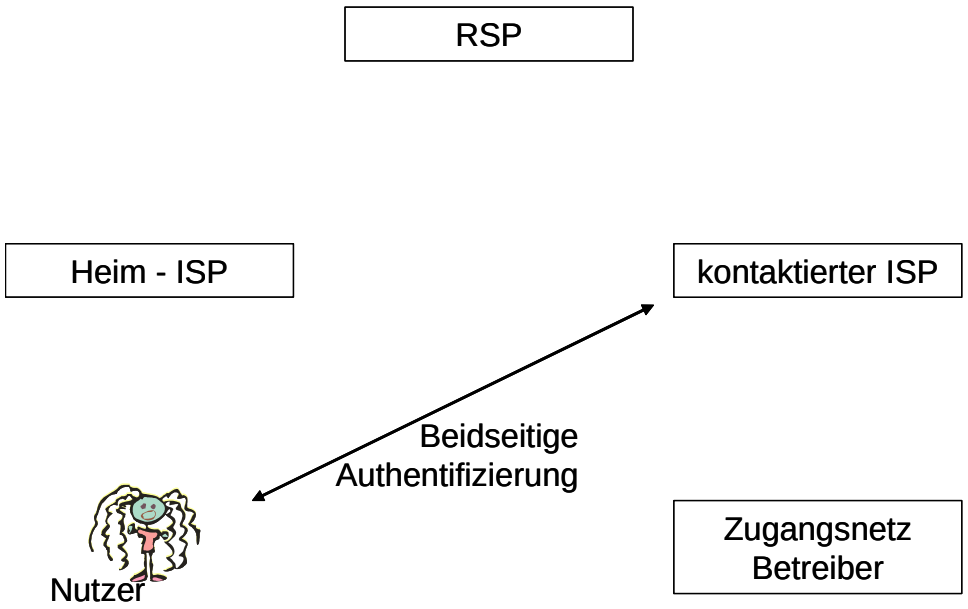


Abbildung 45: Authentifikation ausschließlich durch den kontaktierten ISP

In diesem Szenario müssen andere Parteien wie der RSP oder der Heim-ISP den Endnutzer nicht authentifizieren, wenn der Endnutzer sich mit dem Internet verbindet. Man stelle sich dafür den Fall eines Nutzers vor, der sich mit dem Internet verbindet, um im World Wide Web zu surfen ohne zusätzlich spezifische Ressourcen seines Heim-ISP, wie z. B. Zugang zu seiner E-Mail-Box zu nutzen. Möglicherweise möchte der Endnutzer einige Dienste von dritten Parteien bereitgestellt in der Form von Web-Diensten nutzen, welche Authentifikation benötigen. Diese Art der Authentifizierung liegt außerhalb der hier entwickelten schnellen Authentifikation. Dies kann mit einem Standardprotokoll für Authentifikation, wie z.B. SSL oder TLS gelöst werden.
Unabhängig davon, dass der kontaktierte ISP den Endnutzer authentifizieren will, will umgekehrt eventuell der Endnutzer auch den kontaktierten ISP authentifizieren, da dieser in der Lage ist persönliche Daten des Endnutzers zu sammeln. Daher wird ein Mechanismus zur beidseitigen Authentifikation benötigt.

Der beidseitigen Authentifizierung wird hier ein „challenge-response“ Mechanismus zugrunde gelegt. Im Handshake des Authentifizierungsprotokolls werden Parameter ausgetauscht, welche für die weitere Korrespondenz notwendig sind.

Dieser Mechanismus wird eingeleitet durch den Endnutzer U, indem er eine *hello* Nachricht sendet, welche die Identität des Nutzers U enthält, sowie eine vom Nutzer U generierte Zufallszahl r_U. Nach Erhalt dieser Nachricht, erzeugt B eine Nachricht $cert_B, r_B, U, t, sig_B(r_U, r_B, U, t)$, in der $sig_B(r_U, r_B, U, t)$ B's Signatur eines Hashwertes von r_U, r_B, U, t bedeutet. In dieser Nachricht, repräsentiert r_B einen von B erzeugten Zufallswert, $cert_B$ repräsentiert B's Zertifikat und t ist die Zeit, zu der B die Nachricht

erzeugt hat. U antwortet dann mit der Nachricht $cert_U, B, t, sig_U(r_U, r_B, B, t)$. Hierbei repräsentiert $cert_U$ U's Zertifikat. U sollte die Nachricht nur signieren wenn der Zeitwert von B akzeptabel ist. Wenn dies nicht der Fall ist, dann sollte U stattdessen das Protokoll abbrechen. Bei den Zufallszahlen muss es sich um so genannte „echte" Zufallszahlen handeln. Das bedeutet, dass alle im Wertebereich liegenden Zahlen mit gleicher Wahrscheinlichkeit im Rahmen des Erzeugungsprozesses generiert werden. Das Protokoll ist in Abbildung 46 dargestellt. Der Zweck von t wird weiter unten erklärt.

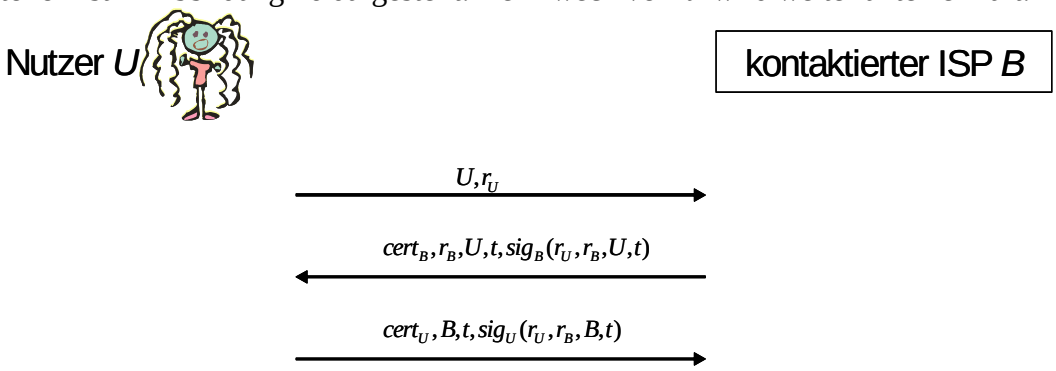


Abbildung 46: Protokoll zur beidseitigen Authentifikation

ISP B speichert die erhaltene Signatur und die zugehörigen Daten in einer Datenbank, um falls nötig in der Lage zu sein, andere Parteien davon zu überzeugen, dass die Authentifikation korrekt ausgeführt wurde. Da der Beweis auch die Zeit t beinhaltet, zu der die Authentifikation durchgeführt wurde, kann er auch für andere Zwecke verwendet werden wie z.B. die Abrechnung. Dafür speichert B die folgenden Werte in seiner Datenbank für jedes Authentifikationsprotokoll, das er abgewickelt hat: $cert_U, r_U, r_B, B, t, sig_U(r_U, r_B, B, t)$.

Falls nötig oder gewünscht, sind so der RSP R oder der Heim-ISP A in der Lage nachzuprüfen, ob B wirklich den Nutzer U authentifiziert hat. Ohne die Zeit t innerhalb des Protokolls, müssten die anderen Parteien die Zufallswerte r_U, r_B speichern, um in der Lage zu sein, Angriffe durch einen betrügerischen B zu erkennen, der alte Signaturen, welche er von U erhalten hat, wiedereinspielt. Wenn der Zeitwert t mitgeschickt wird, können solche Wiedereinspielungs-Angriffe vom RSP R oder ISP A entsprechend erkannt werden.

Wenn der Nutzer nicht in der Lage ist die Signaturen zu überprüfen, die er im Rahmen der beidseitigen Authentifizierung des Protokolls erhalten hat, kann er trotzdem den Rest des Protokolls ausführen. Die Folge davon ist allerdings, dass die Beidseitigkeit der Authentifizierung verloren geht. Dies erfordert zusätzliches Vertrauen des Nutzers bezüglich des korrekten Verhaltens des kontaktierten ISPs.

5.1.1.1.1 Sicherheitsbetrachtung des Authentifikationsprotokolls

In diesem Abschnitt wird die Sicherheit des in Abbildung 46 dargestellten Protokolls dargestellt. Es ist davon auszugehen, dass ein potentieller Angreifer, die Möglichkeit hat, beliebig Nachrichten zu Löschen, zu Verändern oder zu Verfielfältigen.

Nachdem im ersten Schritt der Nutzer seine Identität und als Challenge eine Zufallszahl geschickt hat, erwartet er die Antwort seines Gegenübers. Da diese nochmals dieselbe Zufallszahl und die Identität des Nutzers enthält und diese beiden Werte in eine signierte Prüfsumme eingehen, kann sofort eine Manipulation der ersten Nachricht des Nutzers bzw. eine Manipulation seitens des kontaktierten ISPs vom Nutzer erkannt werden. Die Authentifikation wäre in diesem Fall gescheitert.

Bei korrektem Ablauf der ersten beiden Protokollschritte, d.h. der ersten beiden verschickten Nachrichten, kann der Nutzer sich Dank des Zertifikates und Dank der Signatur des kontaktierten ISP B sicher sein, dass er mit dem „richtigen“ im Zertifikat angegebenen ISP kommuniziert, falls die Überprüfung der Signatur erfolgreich war und ein positives Ergebnis vorgelegen hat, da nur der richtige ISP B diese Signatur erstellen kann, was hier vorauszusetzen ist. Wenn die Signaturüberprüfung kein postivies Ergebnis liefert oder die signierte Prüfsumme nicht korrekt ist, ist der Authentifizierungsvorgang gescheitert, da eine Manipulation vorgelegen haben muss. Der Zeitwert sorgt zusätzlich dafür, dass ein Wiedereinspielen der Nachricht zu einem Späteren Zeitpunkt erkannt wird, was auch zum scheitern des Authentifikationsvorganges führen würde.

Analog zur zweiten Nachricht, sendet der Nutzer an den kontaktierten ISP nun eine dritte Nachricht, die u.a. den vom ISP geschickten Zufallswert und den Zeitwert enthält. Durch die signierte Prüfsumme ist sicher gestellt, dass eine Manipulation der Nachricht erkannt werden kann. Der ISP kann die Signatur des Nutzers anhand des Nutzerzertifikates überprüfen. Falls die Signatur korrekt und das Zertifikat gültig ist, kann er sich der Identität des Nutzers sicher sein.

Ein „Man-in-the-middle“, der versucht einerseits dem Nutzer vorzutäuschen der ISP zu sein und andererseits dem ISP vorzutäuschen der Nutzer zu sein, würde zu einem solchen vorhaben die jeweiligen privaten Schlüssel der beiden benötigen, um die entsprechenden Signaturen erstellen zu können. Solange er über diese nicht verfügt, kann er sich nicht unerkannt zwischen die beiden Kommunikationspartner setzten.

Eine beidseitige Authentifizierung, welche das Protokoll zum Ziel hat, wird also gewährleistet. Die hier gemachten Sicherheitsbetrachtungen gelten auch für die folgenden Fälle.

5.1.1.1.2 Delegation der Authentifikation zum RSP

Wenn der ISP B nicht in der Lage ist die Authentifizierung mit dem entsprechenden Mechanismus durchzuführen, dann ist eine Lösung, die es dem ISP B ermöglicht, die Authentifikation des Nutzers zum RSP R zu delegieren, sinnvoll. In diesem Fall delegiert

B die anfallende Authentifizierungsarbeit an R. Dabei wird vorausgesetzt, dass R in der Lage ist ein entsprechendes Protokoll auszuführen. In der Abbildung 47 sind die Authentifikationsbeziehungen der involvierten Rollen in dem hier zugrunde gelegten Szenario schematisch dargestellt.

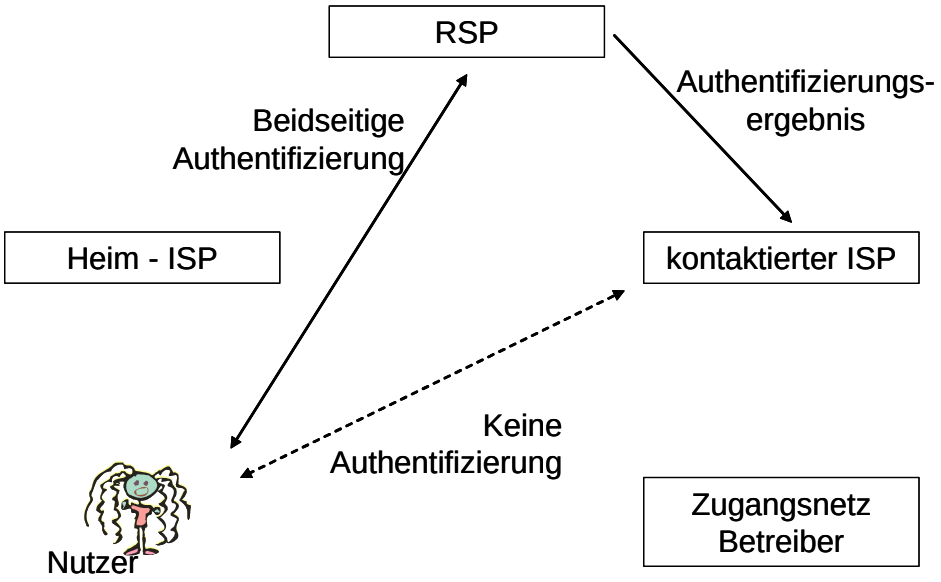


Abbildung 47: Authentifikation durch RSP nach Delegation

Die Lösung für die delegierte Authentifikation basiert grundlegend auf der Lösung ohne Delegation zum RSP. Alle Nachrichten, welche ISP B von Nutzer U erhält, werden zu RSP R weitergeleitet, damit dann R alle notwendigen Berechnungen durchführen bzw. Aktionen ausführen kann. Nachdem R U authentifiziert hat, sendet R eine Nachricht $B, R, U, t, sig'_R(B, R, U, t)$ an B, in welcher er B zusichert, dass die Identität des Nutzers, mit dem er in Verbindung steht, echt ist oder nicht. R nutzt dabei einen Signatur Algorithmus sig'. Dieser Algorithmus wird so ausgewählt, dass die eine Partei B in der Lage ist, ihn zu handhaben. Die Signatur kann von B überprüft werden, um zu sehen mit welcher Partei er kommuniziert. Diese Verifikation basiert auf einer Vertrauensbeziehung zwischen B und R. B sollte die Nachricht, welche er von R erhält speichern, da dies nützlich für die Abrechnung sein kann. Dies ist in folgender Abbildung 48 dargestellt.

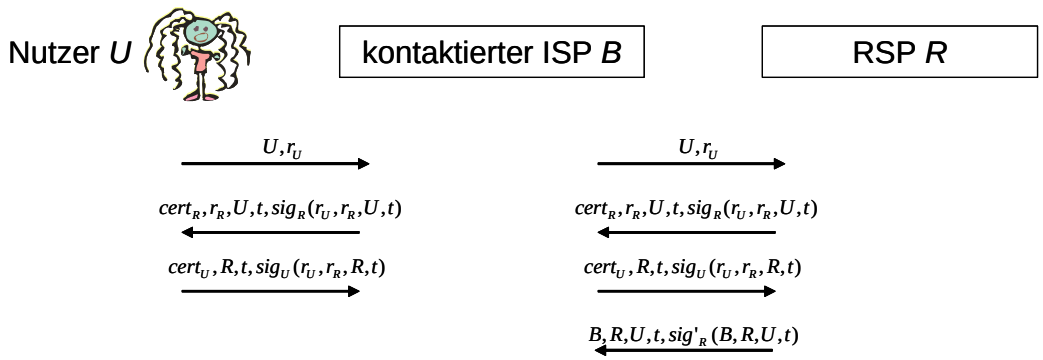


Abbildung 48: Protokoll zur beidseitigen Authentifikation mit Delegation

5.1.1.1.2.1 Sicherheitsbetrachtung

Die Sicherheit der Authentifizierung wird im Delegationsfall genauso gewährleistet, wie im Falle ohne Delegation. Der kontaktierte ISP kann wie ein Angreifer in der Leitung prinzipiell zwar die über ihn geschickten Nachrichten beliebig manipulieren. Eine derartige Manipulation wird durch die im Protokoll enthaltenen signierten Hashwerte jedoch entdeckt, was einen derartigen Angriff auf einen Denial of Service Angriff, der von jedem Angreifer, der die Möglichkeit hat Pakete zu manipulieren geführt werden kann, reduziert. Die letzte Nachricht des Protokolls aus Abbildung 48 sorgt dafür, dass der kontaktierte ISP vom RSP die Bestätigung der Identität des Nutzers erhält. R kann nicht plötzlich eine andere Identität von U vortäuschen, da die erklärte Identität von U dem ISP ja geschickt wurde. Der Zeitwert wurde von B selbst generiert. Daher würde eine Manipulation desselben auch von B bemerkt werden. Was die Überprüfung der Identität angeht, muss B dem RSP R vertrauen, wie in Kapitel 3 bereits ausgeführt. Der RSP kann also prinzipiell einem Nutzer, dessen Identität im Rahmen der Signatur und Zertifikatsüberprüfung nicht bestätigt werden konnte, trotzdem dem ISP B gegenüber bestätigen. Damit würde er sich allerdings selber schaden, da der ISP B ja von ihm auf Basis dieser Bestätigung später für seine Dienste Geld verlangt. Der RSP hat prinzipiell auch die Möglichkeit korrekt identifizierte Nutzer dem ISP gegenüber als nicht identifiziert zu melden. Dies käme einem „Denial of Service“ gegenüber dem Nutzer gleich, womit der RSP sich auch selber Schaden würde, da er letztlich daran verdient, dass er mit seinem Dienst den Nutzern das Roaming ermöglicht. Insofern ist ein Betrug von Seiten des RSP unwahrscheinlich.
Die hier gemachten Sicherheitsbetrachtungen gelten auch für die folgenden Fälle.

5.1.1.2 Authentifikation durch den kontaktierten ISP und den Heim-ISP

Im Gegensatz zum vorigen Abschnitt, wird jetzt von einem Nutzer ausgegangen, der Zugriff auf Ressourcen benötigt, die von seinem Heim-ISP bereit gestellt werden., so z.B. seine Mail Box, welche bei seinem Heim-ISP verwaltet und betrieben wird. In diesem Fall muss der Heim-ISP den Nutzer authentifizieren. Außerdem muss der End-Nutzer ebenso ein Interesse daran haben, dass sein Heim-ISP den Nutzer, der eine Anfrage nach Zugriff auf eine Ressource stellt, sicher korrekt authentifiziert, z.B. aus Gründen der Wahrung der Privatsphäre. Des Weiteren kann der Endnutzer ebenso daran interessiert

sein seinen Heim-ISP zu authentifizieren. Dementsprechend ist eine zusätzliche beidseitige Authentifikation zwischen dem Endutzer und dem Heim-ISP notwendig.

Um dem diesem Abschnitt zugrunde gelegten Szenario gerecht zu werden, müssen wir zwei beidseitige Authentifizierungen zum einen zwischen Endnutzer und kontaktiertem ISP und zum anderen zwischen Endnutzer und Heim-ISP. Für die erste Authentifizierung, können wir genau dasselbe Protokoll wie in Abschnitt 5.1.1.1 beschrieben verwenden. Die beidseitige Authentifizierung zwischen dem Nutzer und dem kontaktierten ISP kann ohne zusätzliche Unterstützung anderer Parteien während des Abwickelns des Protokolls durchgeführt werden. Bezüglich des Protokolls, welches zwischen Nutzer und Heim-ISP zwecks beidseitiger Authentifizierung abgewickelt wird. Prinzipiell kann jedes Protokoll hier eingesetzt werden kann. Dieses Protokoll wird hier nicht näher betrachtet, da es außerhalb des hier zugrunde gelegten Geschäftsmodells liegt.

Man muss an dieser Stelle anmerken, dass diese Lösung zwei Authentifizierungsvorgänge erforderlich macht. Dies kann im Gegensatz zu der Anforderung eines möglichst geringen reduzierten Authentifizierungsaufwandes stehen. Jedoch wird die zweite Authentifizierung nicht in allen Fällen benötigt. Sie ist nur notwendig, wenn der Nutzer Ressourcen benötigt, welche von seinem Heim-ISP bereitgestellt werden.

In der Abbildung 49 sind die Authentifikationsbeziehungen der involvierten Rollen in dem hier zugrunde gelegten Szenario schematisch dargestellt.

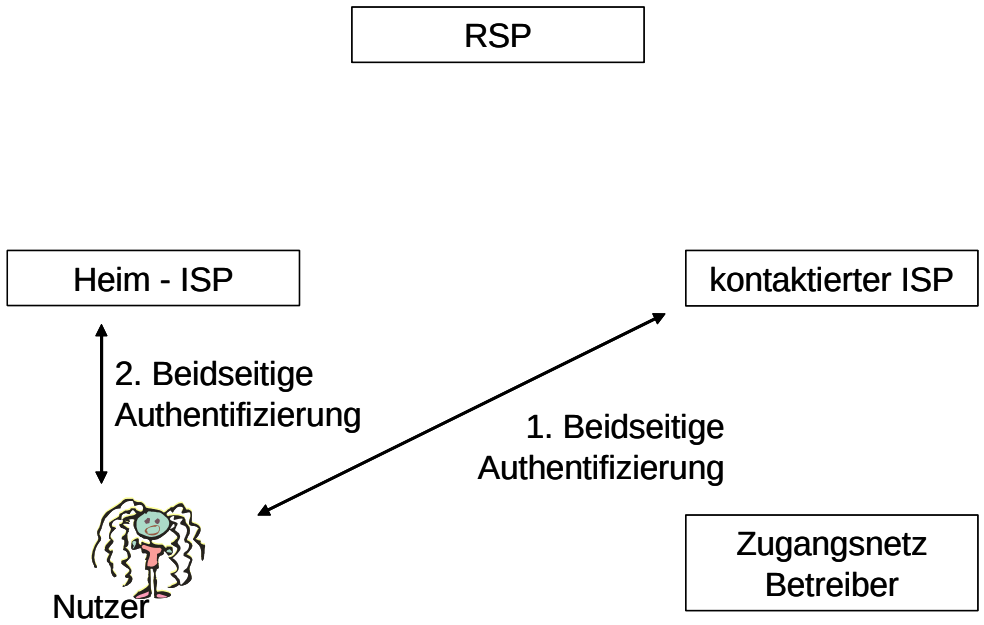


Abbildung 49: Authentifikation durch kontaktierten ISP und Heim – ISP

5.1.1.2.1 Delegation der Authentifikation zum RSP

In diesem Fall entspricht in seinen Abläufen der Authentifizierung zwischen Nutzer und kontaktiertem ISP, die in Abschnitt 5.1.1.1.1 beschrieben ist. Die Authentifizierung zwischen dem Heim-ISP und dem Nutzer ist von der Delegation nicht beeinflusst, d.h. jedes Protokoll für beidseitige Authentifizierung kann angewendet werden, wie in Abschnitt 5.1.1.2 beschrieben. In der Abbildung 50 sind die Authentifikationsbeziehungen der involvierten Rollen in dem hier zugrunde gelegten Szenario schematisch dargestellt.

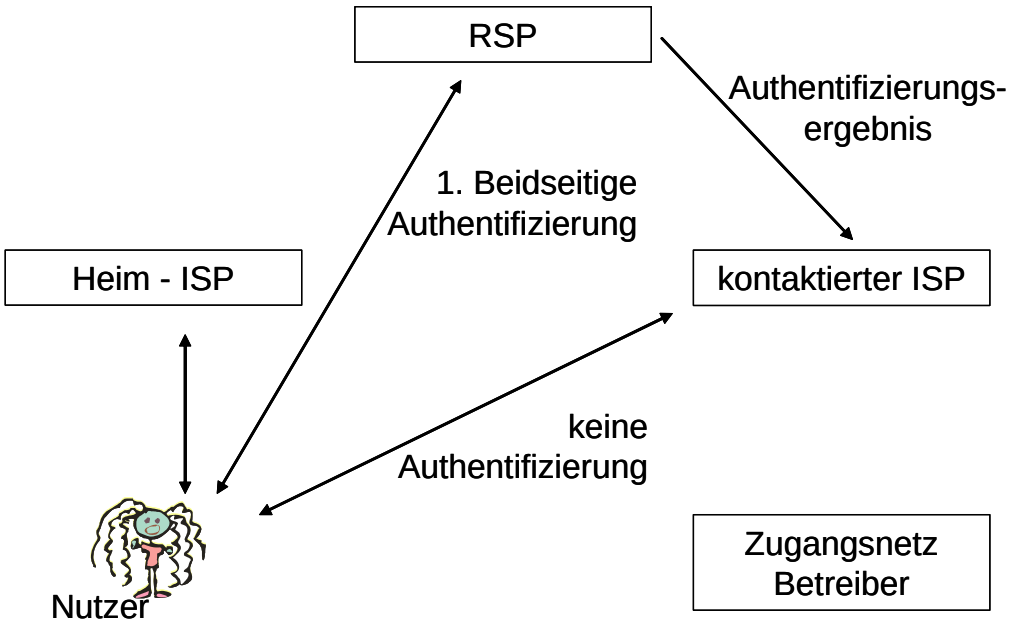


Abbildung 50: Authentifikation durch Heim ISP und RSP nach Delegation

5.1.1.3 Authentifikation durch den kontaktierten ISP und Weiterleitung der Authentifikationsinformation

In diesem Abschnitt gelten die gleichen Voraussetzungen wie in Abschnitt 5.1.1.2. Es wird von einem Szenario ausgegangen, indem ein Nutzer auf Ressourcen zugreifen möchte, die von seinem Heim-ISP direkt bereitgestellt werden, wie z.B. seine persönliche Mailbox, die von seinem Heim-ISP betrieben wird. Es ist hier ebenfalls einleuchtend, dass der Heim-ISP den Nutzer authentifiziert. Anstatt den Endnutzer in eine zweite Authentifizierung zu verwickeln, kann das unten gezeigte Protokoll eingesetzt werden, welches es sowohl dem Heim- und dem kontaktierten ISP als auch dem RSP erlaubt den Endnutzer zu identifizieren ohne zusätzlichen Aufwand für den Nutzer. Genauer gesagt, beinhaltet das Protokoll eine beidseitige Authentifizierung zwischen Nutzer und kontaktiertem ISP und erlaubt zusätzlich dem Heim-ISP und RSP den Nutzer zu authentifizieren. Was den Teil der Authentifikation des Nutzers durch den RSP und den Heim-ISP betrifft, ermöglicht das Protokoll die Identifizierung des Nutzers auf Basis der Daten aus dem Authentifizierungsvorgangs zwischen kontaktiertem ISP und Nutzer. In

der Abbildung 51 sind die Authentifikationsbeziehungen der involvierten Rollen in dem hier zugrunde gelegten Szenario schematisch dargestellt.

In der folgenden Lösung leitet der kontaktierte ISP die Authentifizierungsinformation zum RSP weiter und der RSP leitet sie wiederum noch zum Heim-ISP weiter. Da die Authentifizierungsinformation als Nachweis einsetzbar ist, können der Heim-ISP und der RSP überprüfen, ob der kontaktierte ISP wirklich den entsprechenden Nutzer authentifiziert hat. Die Hauptidee für diese Lösung basiert auf den Protokollen des vorigen Abschnitts. Das spezielle Design des gewünschten Protokolls hängt von Variationen des Vertrauensmodells ab.

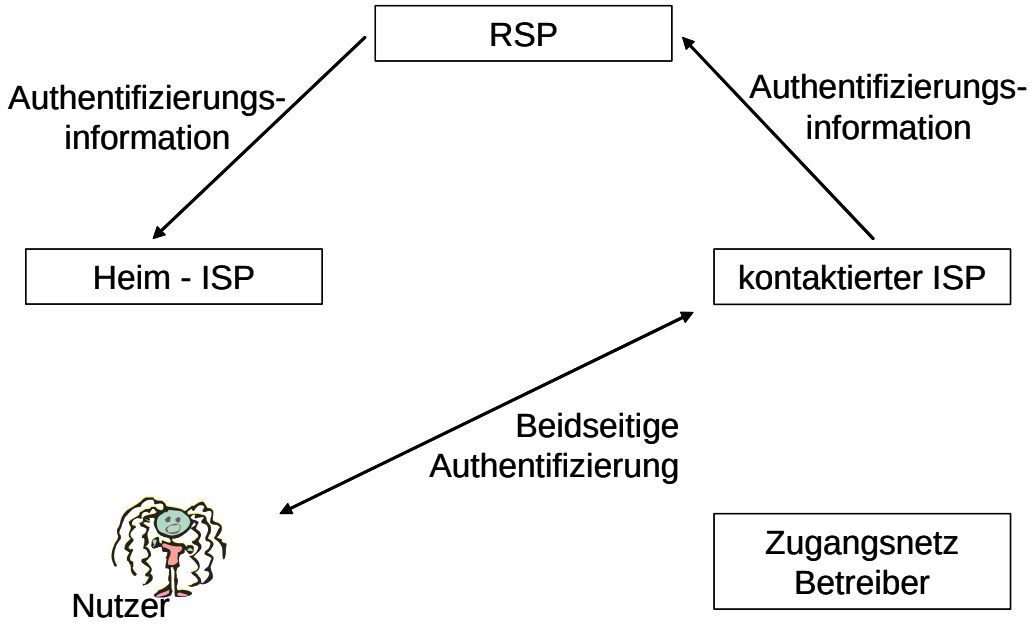


Abbildung 51: Nutzer Authentifikation ohne Zugangsnetzbetreiber

Der Unterschied in den Vertrauensbeziehungen besteht in der Problematik des Heim-ISPs, die gegenwärtige Adresse des Nutzers der Identität des Nutzers zuzuordnen, z.B., die vorübergehende IP Adresse des Endgerätes des Nutzers, welche Verwendung findet, wenn der Endnutzer eine Verbindung zum Heim-ISP etabliert. Es kann mehrere Nutzer geben, die einen Vertrag mit dem Heim-ISP eingegangen sind, und die gleichzeitig die Dienste eines bestimmten kontaktierten ISPs nutzen wollen. Daher benötigt der Heim-ISP eine Methode, um die Identität eines Nutzers seiner gegenwärtigen Adresse zuzuordnen. Der entsprechende Vertrauensaspekt beinhaltet, dass die Netz-Adresse, welche der kontaktierte ISP bereitstellt, wirklich zu der Identität gehört, welche mit der Authentifizierungsinformation weitergeleitet wird.

In dem Fall, dass der Heim-ISP dem kontaktierten ISP vertraut bezüglich dieses Aspektes, kann der kontaktierte ISP den Heim ISP einfach mit der benötigten Zuordnungsinformation versorgen. Er übermittelten Netzadresse. Die beidseitige

Authentifizierung zwischen Nutzer U und kontaktiertem ISP B wird ausgeführt von unter Abwicklung eines „challenge-and-response“ Protokolls wie schon zuvor.

Dieser Mechanismus wird dadurch eingeleitet, dass U eine *hello* Nachricht, welche die Identität des Nutzers U und eine von U erzeugte Zufallszahl r_U beinhaltet, an ISP B schickt. Dieser erzeugt nach Empfang dieser Nachricht eine Nachricht von der Form $cert_B, r_B, U, t, sig_B(r_U, r_B, U, t)$, wobei $sig_B(r_U, r_B, U, t)$ B's Signatur eines Hashwertes, welcher aus r_U, r_B, U, t errechnet wird, darstellt. In dieser Nachricht, repräsentiert r_B den von B erzeugten Zufallswert, $cert_B$ repräsentiert B's Zertifikat, und t ist der Zeitwert wann B diese Nachricht erzeugt. Dann antwortet U mit der Nachricht $cert_U, B, t, sig_U(r_U, r_B, B, t)$, wobei $cert_U$ U's Zertifikat darstellt. Der Vorgang ist in Abbildung 52 dargestellt. Der Nutzer U sollte diese Nachricht nur signieren, wenn der von B gegebene Zeitwert akzeptierbar ist; wenn nicht sollte U diesen zurückweisen und das Protokoll abbrechen. Die Zufallswerte müssen auf sichere Art und Weise erzeugt werden. Der Sinn des Wertes t wird weiter unten erläutert.

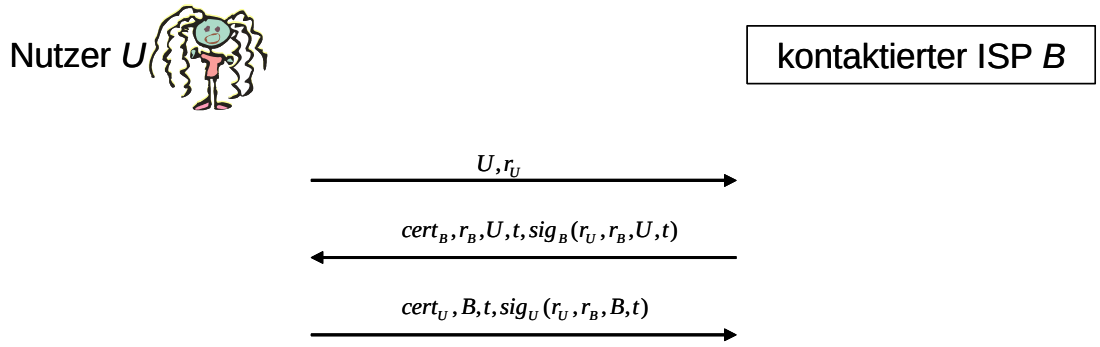


Abbildung 52: Protokoll zur Authentifikation mit vertrauenswürdigem ISP B

ISP B speichert die erhaltenen Signaturen und die zugehörigen Daten in seiner Datenbank, um in der Lage zu sein, die anderen Parteien davon zu überzeugen, dass die Authentifikation korrekt durchgeführt wurde. Da dieser Beweis auch den Zeitpunkt t, zu dem die Authentifizierung durchgeführt wurde, enthält, kann er für spätere Zwecke wie für die Abrechnung eingesetzt werden. Daher speichert B die folgenden Werte für jede von ihm durchgeführte Authentifikation in seiner Datenbank: $cert_U, r_U, r_B, B, t, sig_U(r_U, r_B, B, t)$.

Der Zeitwert t ist jedoch auch aus anderen Gründen im Protokoll enthalten. Andere Parteien können diese Daten zur Authentifizierung eines Nutzers U verwenden, ohne zusätzlich ein weiteres eigenes challenge-and-response Protokoll durchführen zu müssen, welches einen weiteren zeitraubenden Nachrichtenaustausch nötig machen würde. B leitet zwecks Authentifizierung das Ergebnis, welches im Authentifizierungsprotokoll erhalten wurde $cert_U, r_U, r_B, B, t, sig_U(r_U, r_B, B, t)$ zusammen mit der Adressinformation

über das Endgerät des Nutzers ad_U zum RSP R oder ISP A weiter. Falls nötig oder gewünscht sind R oder A in der Lage zu verifizieren, ob B wirklich den Nutzer U authentifiziert hat oder nicht. Ohne den Zeitwert t im Protokoll, müssten die anderen Parteien die Zufallswerte r_U, r_B speichern, um Angriffe von bösartigen ISP B, welche alte von U erhaltene Signaturen erneut einspielen, zu erkennen. Wenn der Zeitwert t zu Verfügung steht, können solche Wiederholungsangriffe jeweils vom RSP R oder ISP A erkannt werden. Die Werte r_u und t können zusammen als Einmalwert – Nonce – angesehen werden. Die Weiterleitung der Authentifikationsinformation ist Abbildung 53 dargestellt.

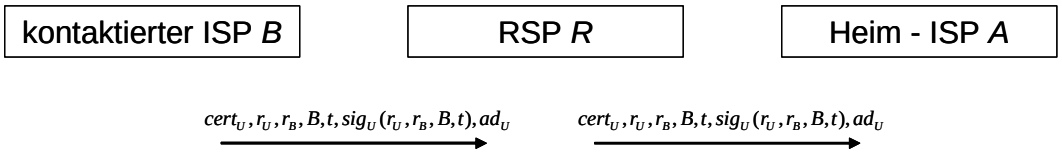


Abbildung 53: Weiterleitung der Authentifikationsinformation bei vertrauenswürdigem ISP B

In dem Fall, dass der Heim-ISP dem kontaktierten ISP bezüglich dieses Aspektes nicht vertraut, muss der kontaktierte ISP den Heim-ISP mit überprüfbarer Zuordnungsinformation versorgen. In der Lösung dieses Falles ist der Endnutzer zusätzlich in die Erzeugung der überprüfbaren Adressinformation involviert. Daher ist das Protokoll für die beidseitige Authentifizierung im Vergleich mit der Lösung des vorherigen Abschnitts 5.1.1.2 modifiziert.

Der Endnutzer U initiiert die Authentifizierung indem er eine *hello* Nachricht, welche sein Identität U und eine Zufallszahl r_U, welche von U erzeugt ist, enthält. Die Notation entspricht der zuvor bereits verwendeten. Nach Empfang dieser Nachricht, erzeugt B eine Nachricht $cert_B, r_B, U, t, sig_B(r_U, r_B, U, t)$. Vorausgesetzt wird hier, dass U die Adresse für sein mobiles Endgerät ad_U von B erhalten hat. Dann antwortet U mit der Nachricht $cert_U, B, t, ad_U, sig_U(r_U, r_B, B, t, ad_U)$. U darf die Nachricht nur signieren, wenn der Zeitwert von B akzeptierbar ist und die gegebene Adresse ad_U gültig ist. Wenn dies nicht der Fall ist, dann muss U das Protokoll abbrechen. Das Protokoll ist in Abbildung 54 dargestellt.

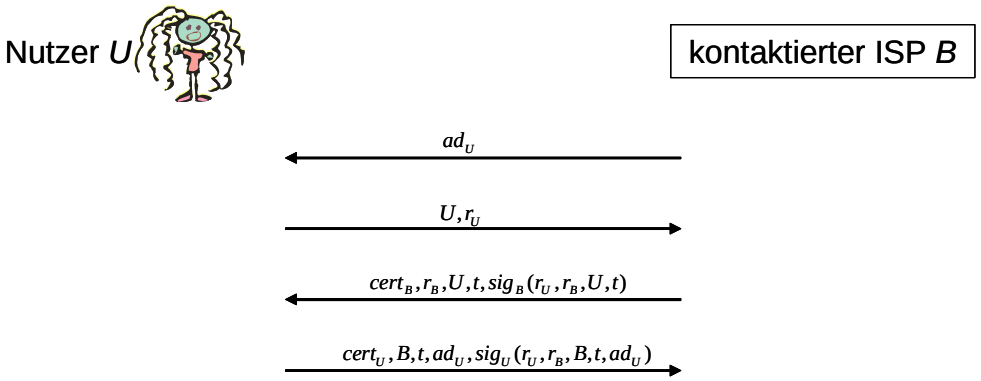


Abbildung 54: Protokoll zur Authentifikation bei nicht vertrauenswürdigem ISP B

Zur Authentifizierung und zur Verifikation der Adresse leitet *B* die Information, welche im Authentifizierungsprotokoll enthalten ist, zum RSP *R* weiter, welcher sie dann wiederum zum ISP *A* weiterleiten kann. Diese Information kann vom RSP *R* und vom ISP *A* dazu verwendet werden, um zu überprüfen, ob *B* wirklich den Nutzer *U* authentifiziert hat und ob die Adresse ad_U von *Us* Endgerät korrekt ist. Der Weiterleitungsvorgang ist in Abbildung 55 dargestellt.

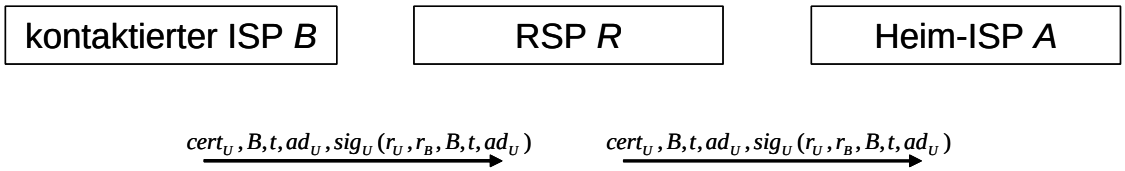


Abbildung 55: Weiterleiten der Authentifikationsinformation im Falle eines nicht vertrauenswürdigen ISP B

5.1.1.3.1 Delegation der Authentifikation zum RSP

Im dem Fall, dass der kontaktierte ISP nicht in der Lage ist, den Nutzer zu authentifizieren, wird die Authentifikation zum RSP delegiert. Danach, leitet der RSP die Information zum Heim-ISP weiter. Dies ist in Abbildung 56 dargestellt.

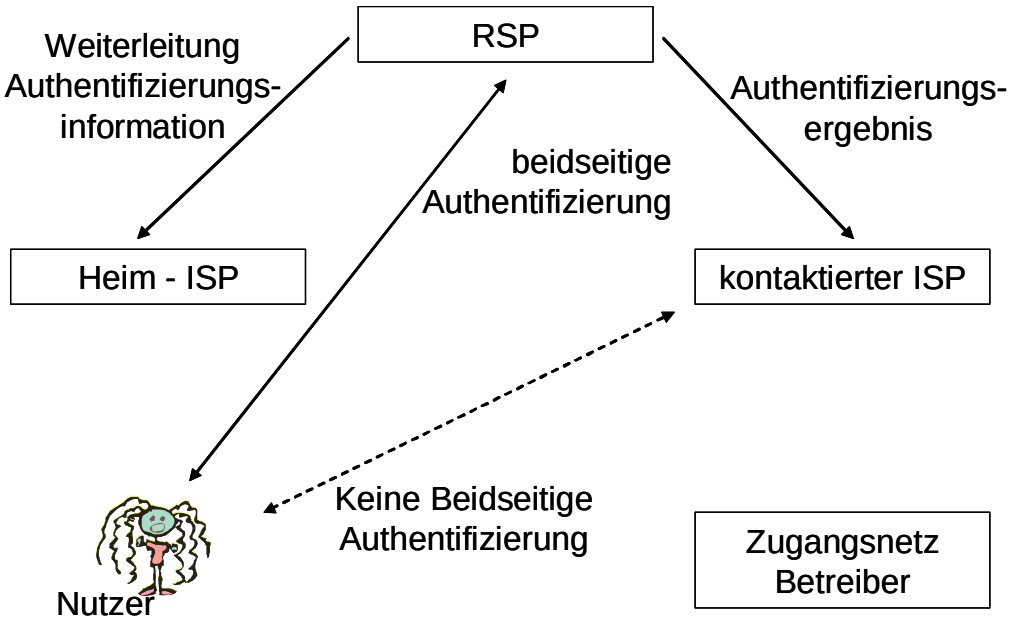


Abbildung 56: Delegation der Nutzer Authentifikation zum RSP

Hier gibt es zwei Fälle zu unterscheiden je nach dem wie vertrauenswürdig der kontaktierte ISP bezüglich der Zuordnung der Nutzeridentität zum Endgerät ist.

Im Falle eines vertrauenswürdigen kontaktierten ISP B muss die Adressinformation nicht zusätzlich geschützt werden. In diesem Fall kann eine ähnliche Lösung wie in Abschnitt 5.1.1.1.2 dargestellt verwendet werden. Der Unterscheid besteht hier darin, dass der RSP R zusätzlich mit der Adressinformation ad_U im Rahmen des Protokolls versorgt werden muss. Wenn der kontaktierte ISP B vertrauenswürdig ist, wird davon ausgegangen, dass er die korrekte Adressinformation ad_U direkt zum RSP R senden kann. Das Authentifikationsprotokoll für diesen Fall ist in Abbildung 57 dargestellt.

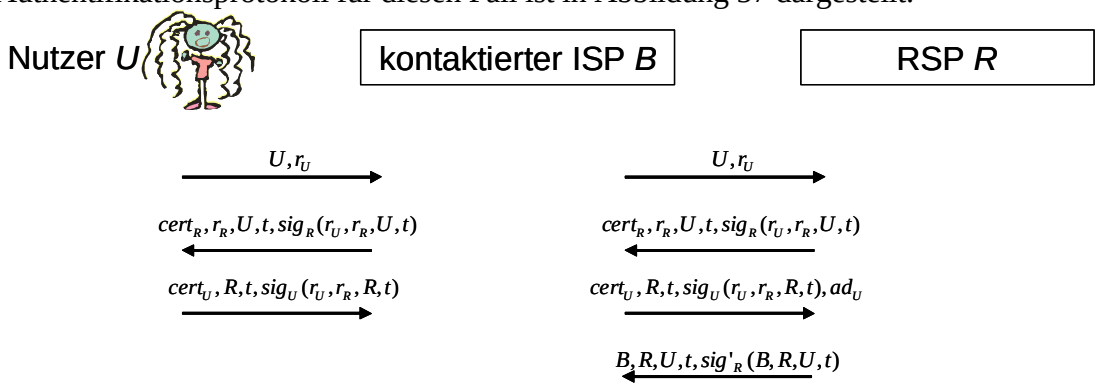


Abbildung 57: Nutzer Authentifikationsprotokoll mit Delegation durch den vertrauenswürdigen kontaktierten ISP

Wenn der kontaktierte ISP nicht vertrauenswürdig ist, dann muss er dem RSP und Heim-ISP mit überprüfbar korrekter Zuordnungsinformation versorgen. In der hier vorgeschlagenen Lösung ist der Nutzer zusätzlich in die Erzeugung der verifizierbaren Adressinformation involviert. Daher ist das Protokoll für die beidseitige Authentifizierung auf passende Weise modifiziert, um die Zuordnungsinformation zu schützen. Wie schon zuvor im Fall ohne Delegation ist die Zuordnungsinformation durch eine Signatur vom Endnutzer geschützt. Das Authentifikationsprotokoll für diesen Fall ist in Abbildung 58 dargestellt.

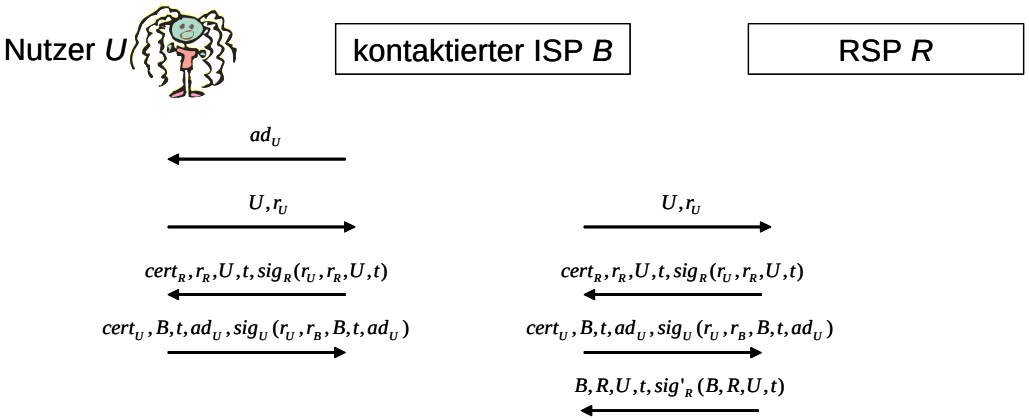


Abbildung 58: Nutzer Authentifikationsprotokoll mit Delegation durch den nicht vertrauenswürdigen kontaktierten ISP

5.1.2 Getrennte Authentifikation durch den Betreiber des Zugangsnetzes und den kontaktierten ISP

Ein typisches Szenario für eine getrennte Authentifikation zwischen Nutzer und Zugangsnetzbetreiber sowie Nutzer und kontaktiertem ISP wäre ein Mobilfunkbetreiber der zunächst unabhängig seine Kunden authentifiziert und danach ein vom Mobilfunkbetreiber unabhängiger ISP der ebenfalls seinen Kunden authentifizieren möchte. Passend dazu wären die Geldflüsse A, D und E aus Kapitel 3.

Im Szenario des vorigen Abschnitts haben der Endnutzer und der Bereitsteller des Zugangsnetzes miteinander Nachrichten ausgetauscht, ohne sich gegenseitig zu authentifizieren. Dort liegt ein Defizit bezüglich des Austausches von authentifizierten Schlüsseln zwischen dem Bereitsteller des Zugangsnetzes und dem Endnutzer für die Verschlüsselung der Luftschnittstelle vor. Dementsprechend werden hier zwei Szenarien vorgeschlagen, in denen der Endnutzer und der Betreiber des Zugangsnetzes sich gegenseitig authentifizieren und zusätzlich der Endnutzer und der kontaktierte ISP eine beidseitige Authentifizierung durchführen. Das grundlegende Ziel kann hierbei auf zwei Arten erreicht werden.

Die erste Möglichkeit wäre, dass zunächst der Endnutzer und der Zugangsnetzbetreiber ein Protokoll zur gegenseitigen Authentifizierung ausführen und danach der Endnutzer und der kontaktierte ISP sich beidseitig authentifizieren. Solch ein Ansatz hat den

Nachteil, dass eine zusätzliche Nutzerauthentifizierung ausgeführt wird, welche negativen Einfluss auf die für den Authentifizierungsvorgang insgesamt benötigte Zeit hat. Auf der anderen Seite existiert ein Standardprotokoll mit dem die gewünschte Funktionalität erreicht werden kann. Wenn man die beidseitige Authentifizierung zwischen Endnutzer und kontaktiertem ISP berücksichtigt, kann man die im vorigen Kapitel 5.1.1 beschriebene Lösung berücksichtigen, welche anderen Parteien erlaubt, wie z.B. andere ISPs und RSPs, den Endnutzer zu authentifizieren.

Eine weitere mögliche Lösung ist es, den Endnutzer, Betreiber des Zugangsnetzes und kontaktierten ISP ein 3-Parteien Authentifizierungsprotokoll abwickeln zu lassen, welches mindestens den Endnutzer und den Zugangsnetzbetreiber beidseitig authentifiziert. Ein derartiges Protokoll existiert, so weit bekannt ist, im Moment nicht.

5.1.2.1 Keine zusätzliche Authentifikation durch den Heim-ISP

Die in diesem Abschnitt beschriebene Lösung ist der in Abschnitt 5.1.1.1 vorgestellten ähnlichen. Der Unterschied zu der dort vorgestellten Lösung besteht darin, dass es eine zusätzliche vom Betreiber des Zugangsnetzes durchgeführte Nutzerauthentifizierung gibt. Als Konsequenz daraus ergibt sich, dass zumindest für die korrespondierenden Parteien dieselbe Lösung wie zuvor verwendet werden kann. In der Abbildung 59 sind die Authentifikationsbeziehungen der involvierten Parteien in dem hier zugrunde gelegten Szenario schematisch dargestellt

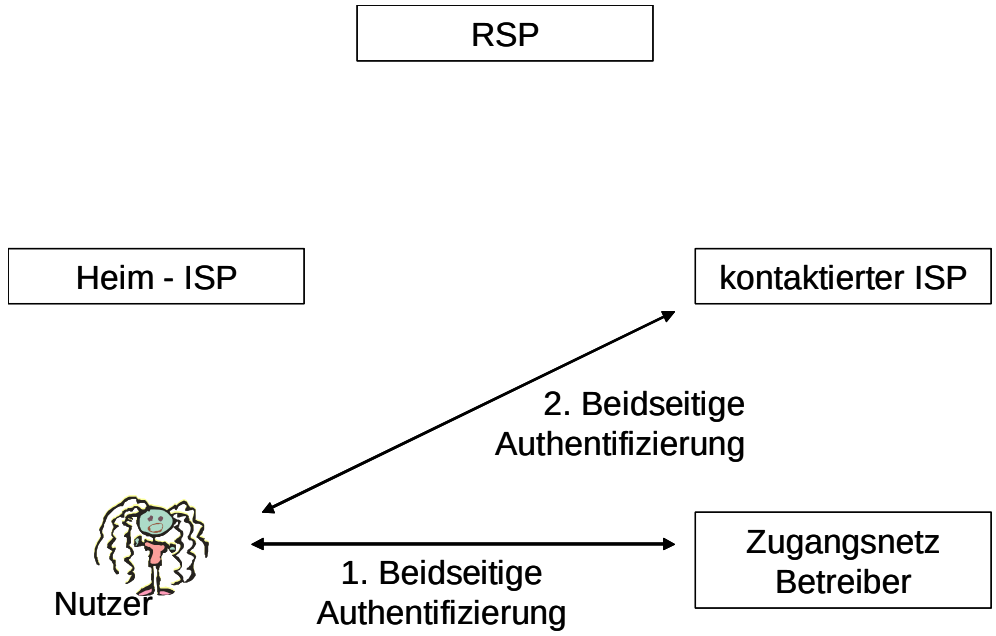


Abbildung 59: Beidseitige Nutzer Authentifikation durch AN Betreiber und kontaktierten ISP

Das bedeutet, dass die Lösung für diesen Fall im Wesentlichen der Lösung von Abschnitt 5.1.1.1 entspricht mit einer zusätzlichen beidseitigen Authentifizierung des

Zugangsnetzebetreibers. Bei der Authentifikation des Zugangsnetzbetreibers handelt es sich je nach zugrunde gelegter Technologie des Zugangsnetzes typischer Weise um eine der in Kapitel 2.2 beschriebenen Protokolle. Man könnte hierfür prinzipiell jedoch auch das hier beschriebene Protokoll zur beidseitigen Authentifizierung aus Abbildung 46 einsetzen.

5.1.2.1.1 Delegation der Authentifikation vom kontaktierten ISP zum RSP

Die folgende Lösung beinhaltet die Delegation der Authentifizierung vom ISP B zum RSP R. Dies kann nötig sein, wenn B nicht in der Lage ist, einen entsprechenden Authentifizierungsmechanismus einzusetzen. In diesem Fall delegiert B die Authentifizierung zu R. Dabei wird vorausgesetzt, dass R in der Lage ist ein geeignetes Protokoll abzuwickeln. Der Fall ist in Abbildung 60 gezeigt.

Endnutzer und Betreiber des Zugangsnetzes führen zuerst ein Protokoll zur beidseitigen Authentifizierung aus, wonach auch der Endnutzer und der kontaktierte ISP Protokolle zur beidseitigen Authentifizierung ausführen. Vorausgesetzt die beidseitige Authentifizierung zwischen Endnutzer und kontaktiertem ISP ist nicht durchführbar, wird die diese vom ISP zum RSP delegiert. Im Falle einer solchen delegierten Authentifizierung vom ISP zum RSP kann man dieselbe Lösung, wie für die Delegation der Nutzerauthentifikation des Zugangsnetzbetreibers anwenden. Die beidseitige Authentifizierung zwischen Zugangsnetzbetreiber und Endnutzer ist davon unberührt.

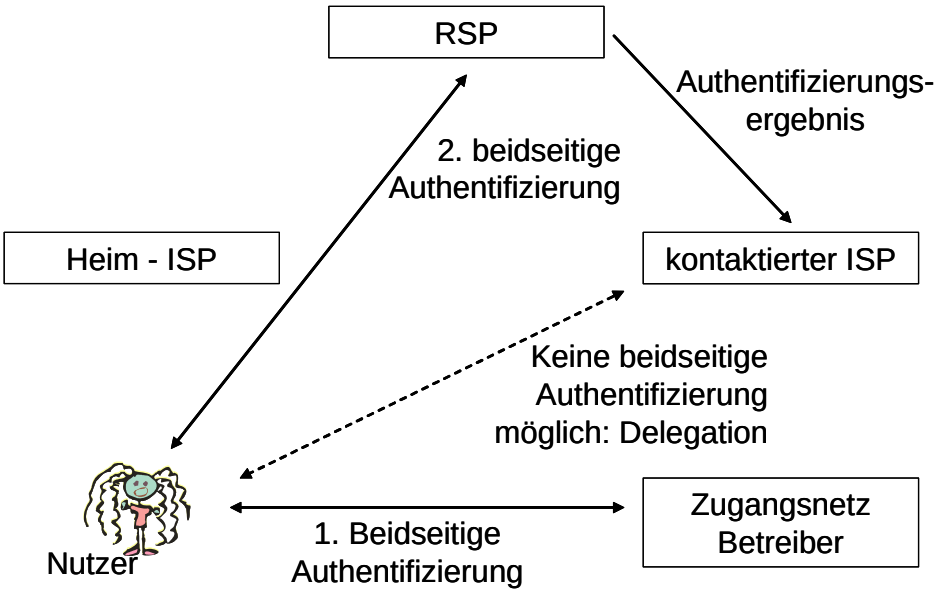


Abbildung 60: Beidseitige Nutzer Authentifikation durch AN Betreiber und kontaktierten ISP mit Delegation zum RSP

5.1.2.2 Zusätzliche Authentifikation durch den Heim-ISP

In diesem Abschnitt wird das Szenario eines Endnutzers analog zu Abschnitt 5.1.2.2 betrachtet, der zusätzlich den Bereitsteller des Zugangsnetzes authentifizieren muss bzw.

umgekehrt. Dies ist in Abbildung 61 dargestellt. Das bedeutet, dass hier dieselbe Lösung wie in Abschnitt 5.1.2.2 mit dem Unterschied einer zusätzlichen beidseitigen Authentifizierung des Zugangsnetzbetreibers angewandt werden kann.

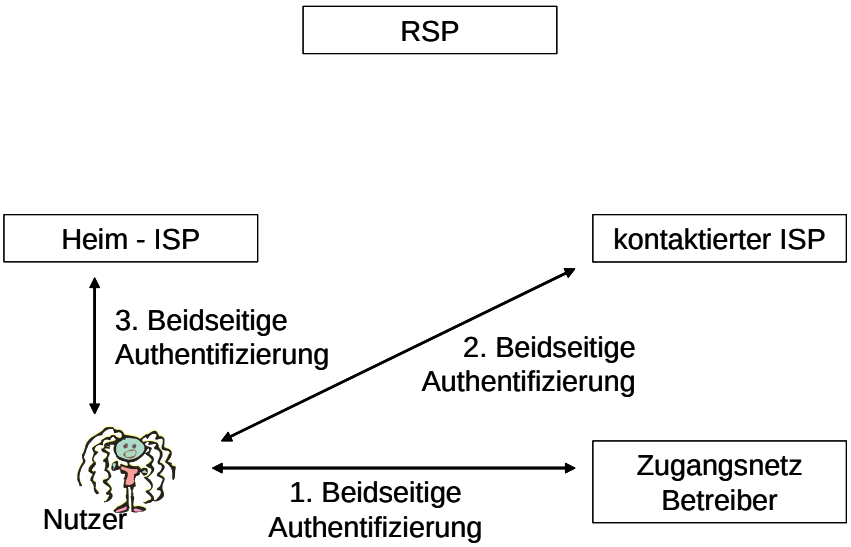


Abbildung 61: Beidseitige Nutzer Authentifikation mit AN Betreiber, kontaktiertem ISP und Heim- ISP

5.1.2.2.1 Delegation der Authentifikation vom kontaktierten ISP zum RSP

Diese Lösung entspricht der Lösung aus Abschnitt 5.1.1.2.1 abgesehen von der Tatsache, dass hier eine zusätzliche Authentifizierung des Zugangsnetzbetreibers durchgeführt wird. Dies ist in Abbildung 62 dargestellt.

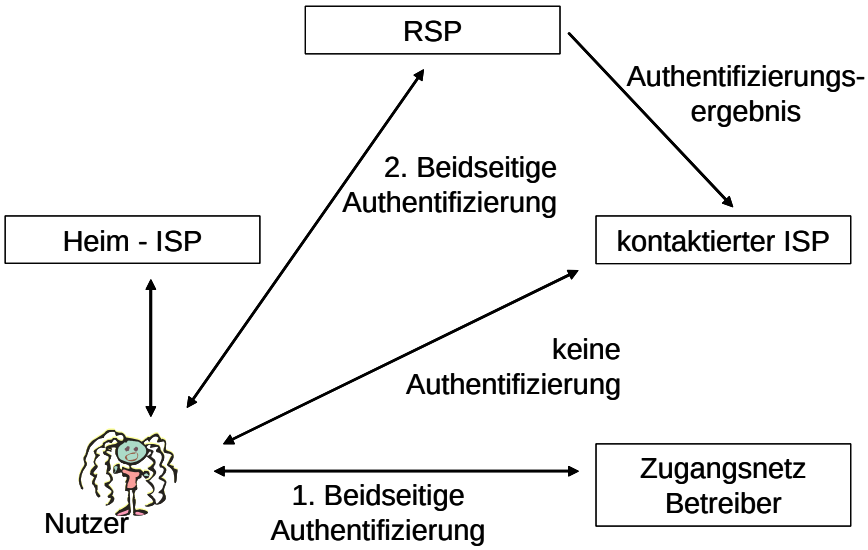


Abbildung 62: Beidseitige Nutzer Authentifikation mit AN Betreiber, kontaktiert ISP, und Heim-ISP mit Delegation zum RSP

5.1.2.3 Authentifikation durch den kontaktierten ISP und Weiterleitung der Authentifikationsinformation

Die Lösung in diesem Abschnitt entspricht prinzipiell der Lösung aus Abschnitt 5.1.1.3 mit dem Unterschied, dass eine zusätzliche Authentifizierung des Zugangsnetzbetreibers durchgeführt wird. Diese Lösung lässt sich für beide Vertrauensmodelle hinsichtlich des Vertrauens zum ISP einsetzen. Sie ist in Abbildung 63 dargestellt.

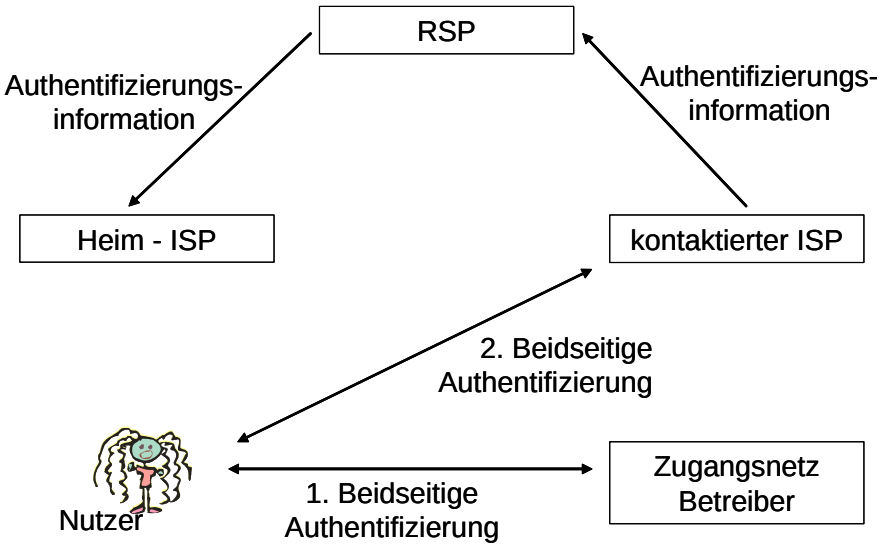


Abbildung 63: Beidseitige Nutzer Authentifikation mit AN Betreiber und kontaktiertem ISP mit Weiterleitung der Authentifikationsinformation

5.1.2.3.1 Delegation der Authentifikation vom kontaktierten ISP zum RSP

Die Lösung in diesem Abschnitt entspricht prinzipiell der Lösung aus Abschnitt 5.1.1.3.1 mit dem Unterschied, dass eine zusätzliche Authentifizierung des Zugangsnetzbetreibers durchgeführt wird. Diese Lösung lässt sich für beide Vertrauensmodelle hinsichtlich des Vertrauens zum ISP einsetzen. Sie ist in Abbildung 64 dargestellt.

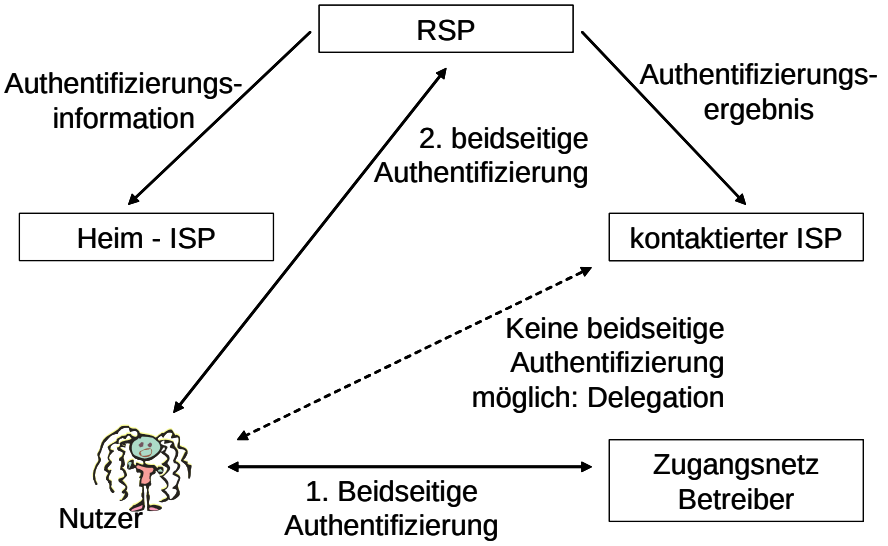


Abbildung 64: Beidseitige Authentifikation von Nutzer und AN Betreiber bzw. kontaktiertem ISP mit Weiterleitung der Authentifikationsinformation und Delegation

5.1.3 Authentifikation durch den Betreiber des Zugangsnetzes und Weiterleitung des Ergebnisses zum kontaktierten ISP

Ein Beispielszenario für den hier dargestellten Fall wäre die Authentifizierung eines Nutzers durch ein Unternehmen, welches sowohl die Rolle eines Zugangsnetzbetreibers als auch die eines ISPs inne hat bzw. bei dem diese beiden miteinander assoziiert sind. Hierzu sind die Geldflüsse D und E aus Kapitel 3 passend.

Der in diesem Abschnitt vorgestellte Fall erlaubt die beidseitige Authentifizierung zwischen Betreiber des Zugangsnetzes und Endnutzer. Die Authentifizierungsinformation wird vom Betreiber des Zugangsnetzes gesammelt und zum kontaktierten ISP weitergeleitet. Diese Information wird dann den anderen Parteien - Heim-ISP, RSP - zum Zwecke der Authentifizierung des Endnutzers zu Verfügung gestellt. Die Lösung ist der in Abschnitt 5.1.1 dargestellten ähnlich. Der Unterschied besteht darin, dass anstatt des kontaktierten ISPs B der Betreiber des Zugangsnetzes W in dieser Lösung mit dem Endnutzer ein Protokoll zur beidseitigen Authentifizierung abwickelt.

Der Fall, dass der Zugangsnetzbetreiber die einzige Partei ist, welche den Endnutzer authentifiziert, ohne dass eine Weiterleitung des Ergebnisses zum kontaktierten ISP erfolgt, ist in der Realität heute häufig der Fall. Da ein solcher mit niemandem kooperierender Zugangsnetzbetreiber kein „Roaming“, so wie in dieser Arbeit verstanden, ermöglicht, wird er hier nicht betrachtet. Eine Weiterentwicklung, die dies ermöglicht, ist der hier betrachtete Fall der Weiterleitung des Ergebnisses zum kontaktierten ISP.

5.1.3.1 Keine zusätzliche Authentifikation durch den Heim-ISP

Die im Folgenden beschriebene Lösung ähnelt der Lösung aus Abschnitt 5.1.1.1. Sie ist schematisch in Abbildung 65 dargestellt.

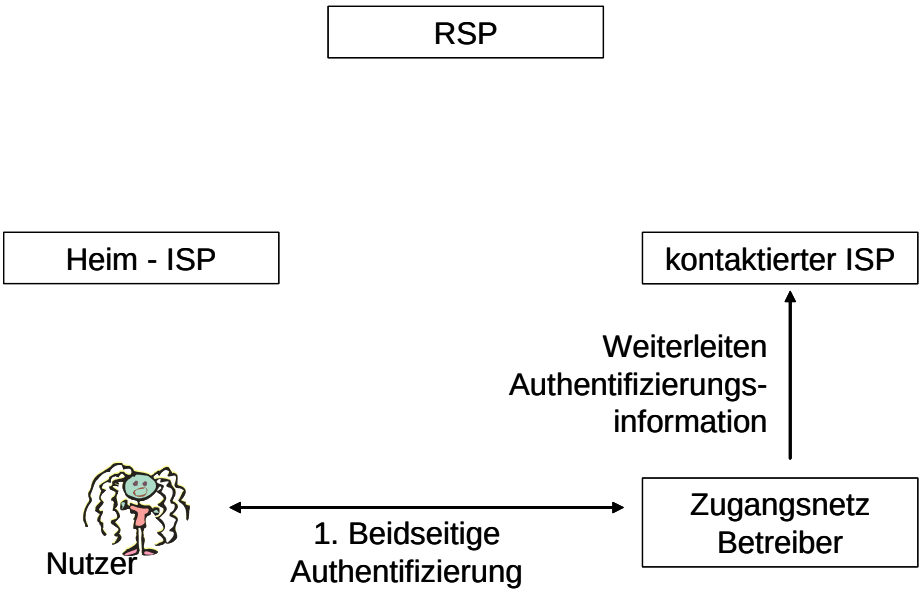


Abbildung 65: Authentifikation durch AN Betreiber und keine zusätzliche Authentifikation durch Heim- ISP

Der Unterscheid besteht darin, dass zum einen der Zugangsnetzbetreiber und nicht der kontaktierte ISP den Nutzer authentifiziert bzw. von diesem authentifiziert wird und zum anderen wird die Authentifizierungsinformation vom Zugangsnetzbetreiber zum kontaktierten ISP weitergeleitet.

5.1.3.1.1 Delegation der Authentifikation zum RSP

In diesem Abschnitt wird der Fall der Delegation der Authentifizierung des Endnutzers vom Zugangsnetzbetreiber zum RSP beschrieben. Hierbei sind zwei Fälle zu unterscheiden. Im ersten Fall, ist der Zugangsnetzbetreiber nicht in der Lage ein entsprechendes Authentifizierungsprotokoll abzuwickeln. Daher delegiert er die Authentifizierung des Nutzers zu dem ihm assoziierten ISP. Im zweiten Fall ist der kontaktierte ISP nicht in der Lage die zu ihm weitergeleitete Authentifizierungsinformation zu verifizieren und delegiert daher die Verifikation dieser Information:

Im ersten Fall entspricht die Lösung zum großen Teil der der Lösung aus Abschnitt 5.1.3.1.1. Der Unterschied besteht darin, dass hier der Betreiber des Zugangsnetzes die Authentifizierung des Nutzers zum ISP delegiert. Dies ist in Abbildung 66 dargestellt. Es besteht weiterhin die Möglichkeit für den ISP, die zu ihm delegierte Authentifizierung weiter zum RSP zu delegieren. Darauf wird hier nicht mehr weiter eingegangen.

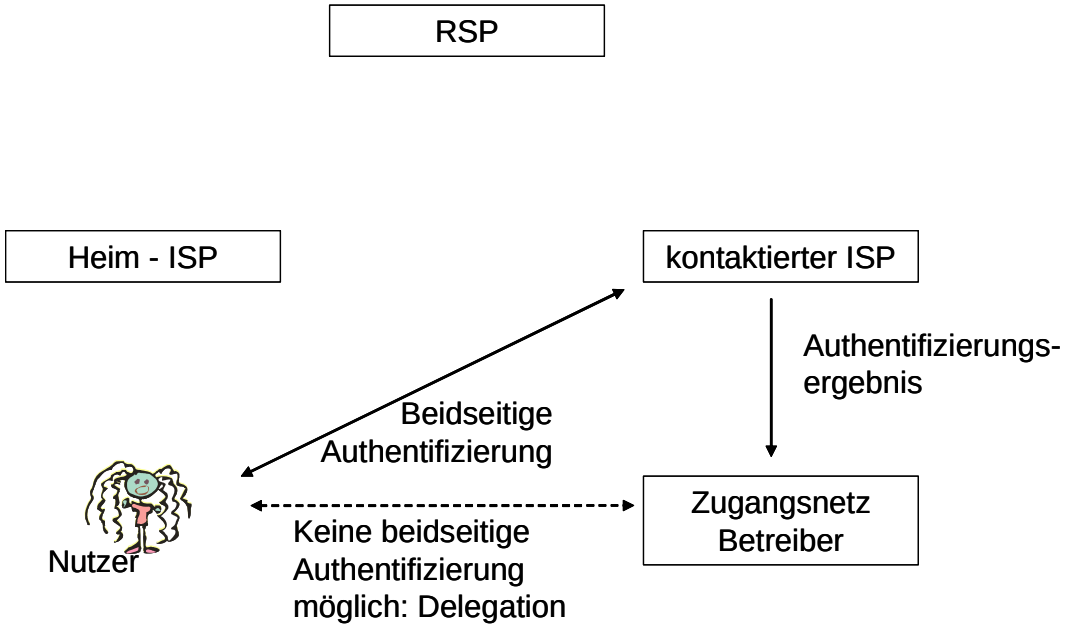


Abbildung 66: Delegation der Authentifikation vom AN Betreiber zum kontaktierten ISP

Im zweiten Fall ist der kontaktierte ISP nicht in der Lage die weitergeleitete Authentifizierungsinformation zu verifizieren und delegiert daher diese Arbeit an den RSP. Dies ist in Abbildung 67 dargestellt.

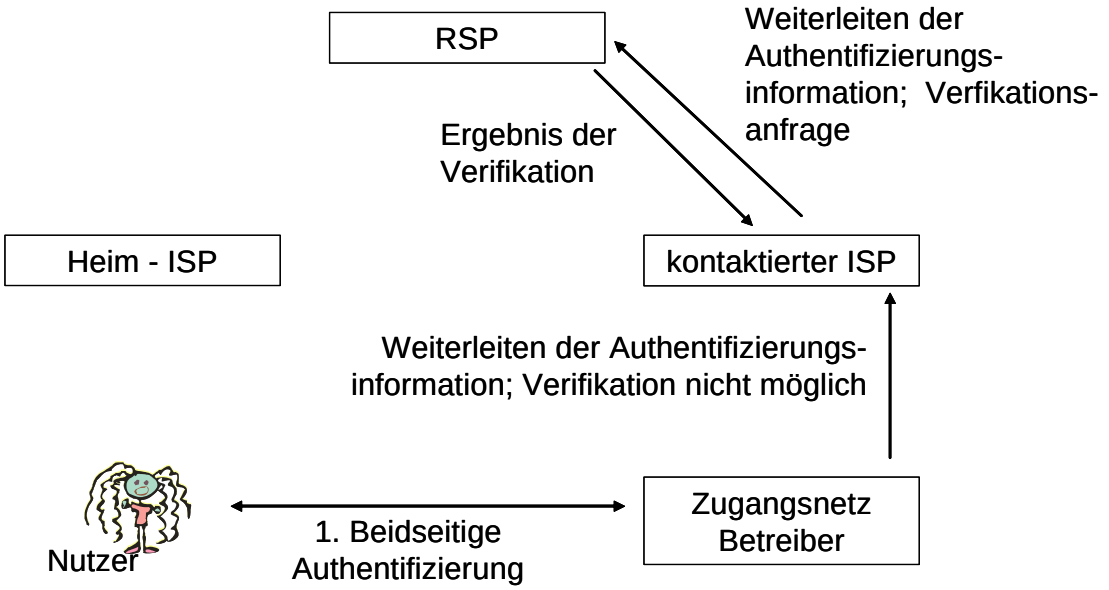


Abbildung 67: Delegation der Verifikation der Authentifikationsinformation

5.1.3.2 Zusätzliche Authentifikation durch den Heim-ISP

Die hier dargestellte Lösung ist der Lösung aus Abschnitt 5.1.1.2 ähnlich. Der Unterschied besteht darin, dass anstelle des kontaktierten ISP *B* in der dargestellten Lösung der Zugangsnetzbetreiber *W* den Nutzer authentifiziert und dass der

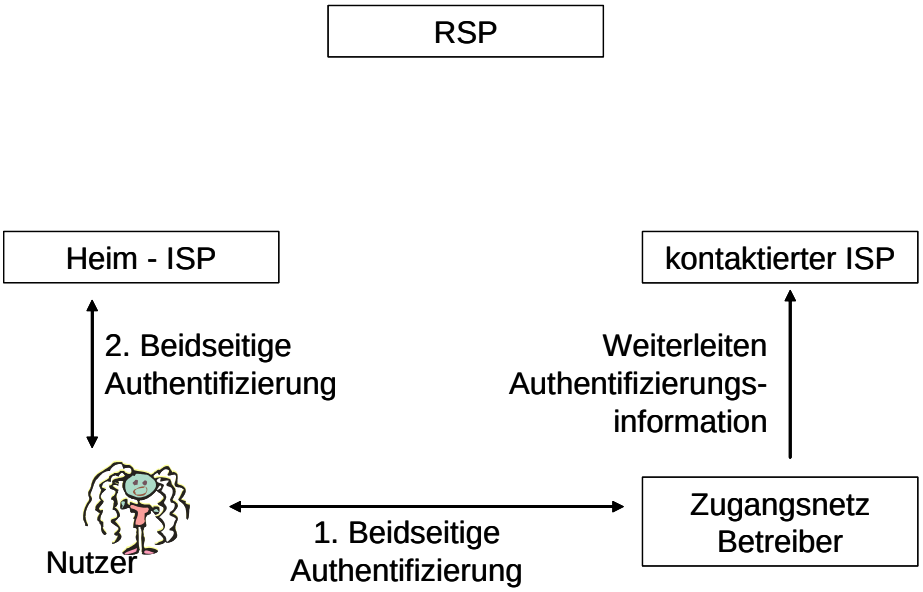


Abbildung 68: Authentifikation durch AN Betreiber und zusätzliche Authentifikation durch Heim- ISP

Zugangsnetzbetreiber die Authentifizierungsinformation zum kontaktierten ISP weiterleitet. Dies ist in Abbildung 68 dargestellt.

5.1.3.2.1 Delegation der Authentifikation zum RSP

In diesem Abschnitt müssen dieselben zwei Fälle wie schon in Abschnitt 5.1.3.1.1 berücksichtigt werden. Im ersten Fall ist der Zugangsnetzbetreiber nicht in der Lage, ein passendes Authentifizierungsprotokoll abzuwickeln und delegiert daher die Authentifizierung zum kontaktierten ISP. Im zweiten Fall ist der kontaktierte ISP nicht in der Lage, die weitergeleitete Authentifizierungsinformation zu verifizieren und delegiert daher die Verifikation. Der erste Fall wird in Abbildung 69 und der zweite Fall in Abbildung 70 dargestellt.

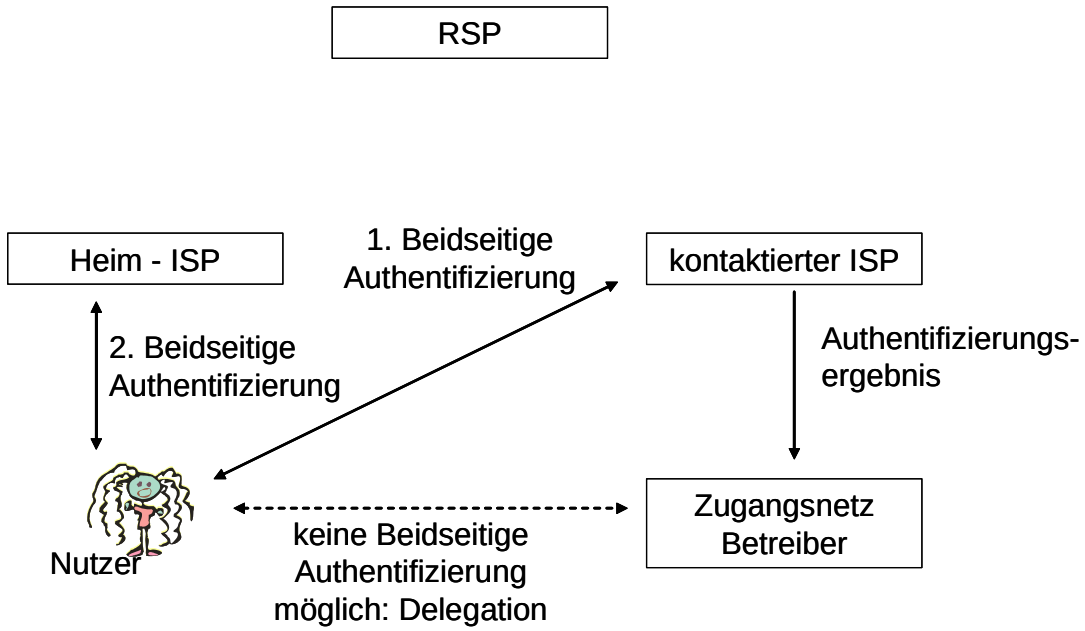


Abbildung 69: Delegation der Authentifikation vom AN Betreiber zum kontaktierten ISP

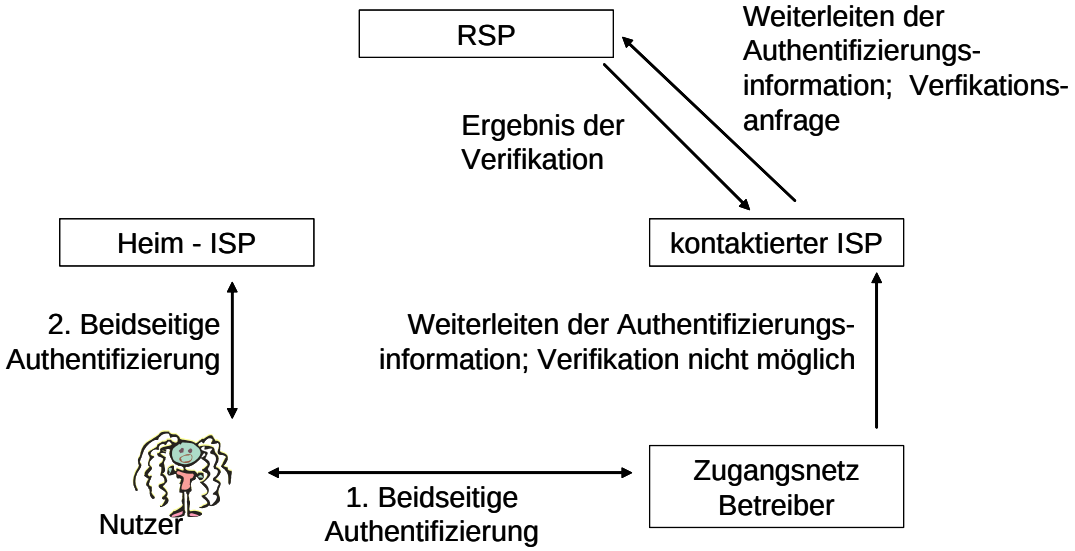


Abbildung 70: Delegation der Verifikation der Authentifikationsinformation

5.1.3.3 Authentifikation durch den Betreiber des Zugangsnetzes und Weiterleitung der Authentifikationsinformation vom kontaktierten ISP zum Heim-ISP

Diese Lösung ist der in Abschnitt 5.1.1.3 gezeigten Lösung ähnlich. Der Unterschied besteht darin, dass zum einen der Zugangsnetzbetreiber *W* und nicht der kontaktierte ISP

B mit dem Endnutzer ein Protokoll zur beidseitigen Authentifizierung abwickelt. und zum anderen darin, dass Zugangsnetzbetreiber die Authentifizierungsinformation des Nutzers zum kontaktierten ISP weiterleitet, wonach der kontaktierte ISP die Authentifizierungsinformation zum RSP und der RSP diese Information zum Heim-ISP des Nutzers weiterleitet. Dies ist in Abbildung 71 dargestellt.

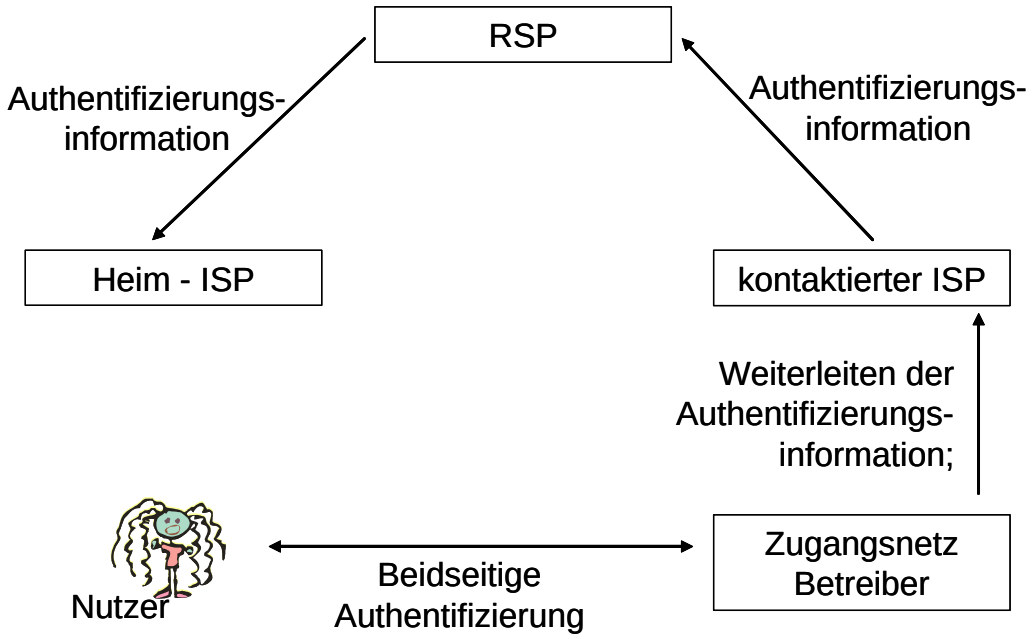


Abbildung 71: Weiterleitung der Authentifikationsinformation durch AN Betreiber

5.1.3.3.1 Delegation der Authentifikation zum RSP

In diesem Abschnitt müssen dieselben beiden Fälle unterschieden werden wie schon in Abschnitt 5.1.3.1.1. Im ersten Fall ist der Zugangsnetzbetreiber nicht in der Lage, zusammen mit dem Nutzer ein entsprechendes Authentifizierungsprotokoll abzuwickeln. Im zweiten Fall ist der kontaktierte ISP nicht in der Lage die an ihn weitergeleitete Authentifizierungsinformation zu überprüfen und Delegiert daher die Verifikation der an ihn weitergeleiteten Authentifizierungsinformation an ihn. Dieser Fall der Re-Delegation wird hier nicht im Detail betrachtet. Der erste Fall ist in Abbildung 72 und der zweite Fall in Abbildung 73 dargestellt.

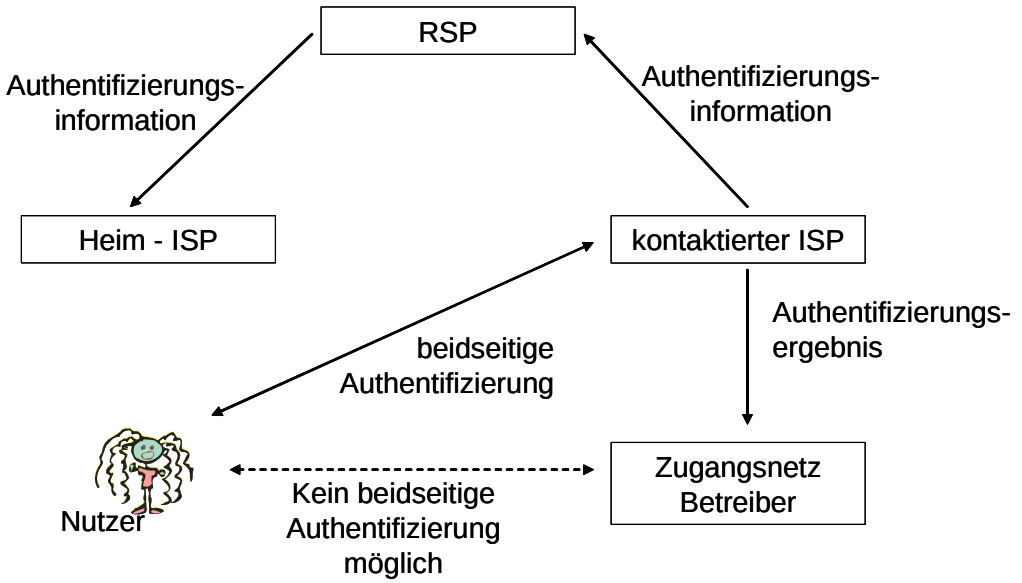


Abbildung 72: Delegation der Authentifikation vom AN Betreiber zum ISP

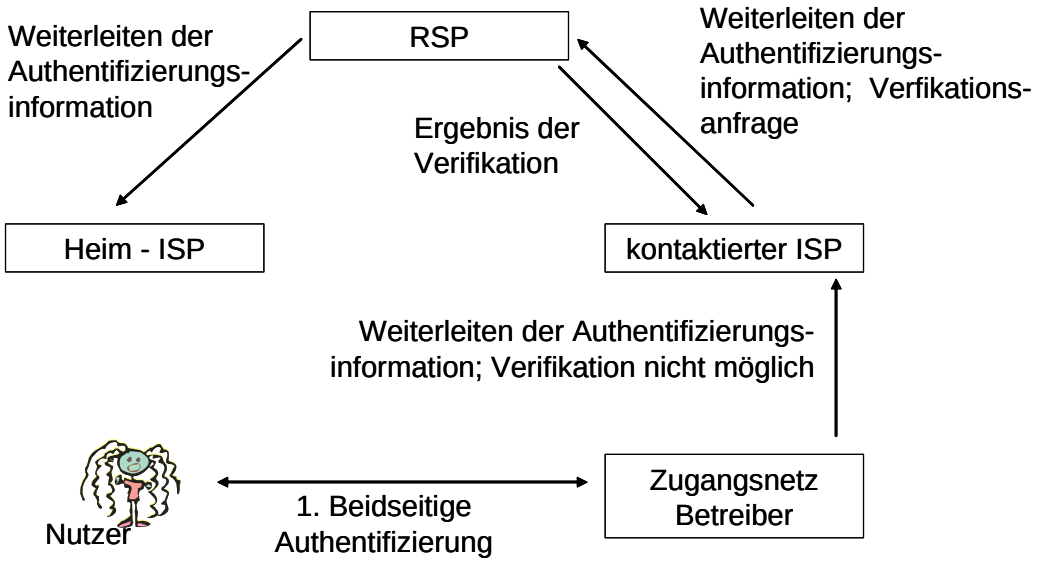


Abbildung 73: Delegation der Verifikation der Authentifikationsinformation

5.2 Roaming mit VPN Zugang Modell

Im Folgenden wird die Lösung für das Modell der Authentifizierung für Roaming mit VPN Zugang präsentiert. Das Ziel dieses Dienstes ist es, den Endnutzer, welcher der Mitarbeiter eines Unternehmens ist, mit einer sicheren VPN Verbindung zum Netz seines Unternehmens zu versorgen. In diesem Modell gehört es zu den Aufgaben des RSPs, den Endnutzer und das Unternehmen mit einem geheimen Schlüssel zu versorgen, auf Basis dessen die VPN Verbindung etabliert werden kann. Dafür muss der RSP sowohl den Endnutzer als auch das Unternehmen authentifizieren. Dabei lassen sich die folgenden Fälle unterscheiden:

- Authentifizierung durch den kontaktierten ISP ohne zusätzliche Authentifizierung durch den Zugangsnetzbetreiber
    - und ggf. Delegation der Authentifizierung zum RSP
- Getrennte Authentifizierung des Nutzers durch den Zugangsnetzbetreiber und den kontaktierten ISP
    - und ggf. Delegation der Authentifizierung zum RSP
- Authentifizierung durch den Zugangsnetzbetreiber und Weiterleitung des Authentifizierungsergebnisses zum kontaktierten ISP und RSP
    - und ggf. Delegation der Authentifizierung zum RSP

Im Rahmen dieser Struktur muss eine zusätzliche beidseitige Authentifizierung von Nutzer und Unternehmen nicht durchgeführt werden. Das liegt daran, dass eine derartige Authentifizierung nach Etablierung der VPN Verbindung automatisch immer möglich ist. Dementsprechend ist hier in der abstrakten Lösung eine zusätzliche beidseitige Authentifizierung unnötig.

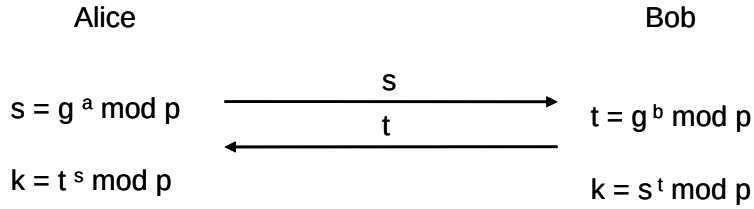
Natürlich besteht die Möglichkeit, dass ein Unternehmen die Dienste eines weiteren ISPs in Anspruch nimmt, um Zugang zum Internet zu erhalten, oder dass es selbst die Rolle eines ISPs spielt. Im Folgenden wird davon ausgegangen, dass das Unternehmen sein eigenes Internet Gateway betreibt und kein zusätzlicher ISP berücksichtigt werden muss.

5.2.1 Authentifikation durch den kontaktierten ISP ohne zusätzliche Authentifikation durch den Betreiber des Zugangsnetzes

In diesem Abschnitt wird das Szenario eines reisenden Angestellten zugrunde gelegt, der eine VPN Verbindung zu seinem Unternehmen benötigt. Wenn wir eine Verbindung zum Internet über einen beliebigen Access Point herstellen, dann wird der kontaktierte ISP den Mitarbeiter zumindest aus Gründen der Autorisierung und Abrechnung authentifizieren wollen. Da der RSP in die Etablierung von sicheren VPN Verbindungen involviert ist, leitet der kontaktierte ISP die im Rahmen des mit dem Nutzer abgewickelten beidseitigen Authentifizierungsprotokolls erhaltene Information an den RSP weiter. Vorausgesetzt, dass diese weitergeleiteten Authentifizierungsdaten die Information beinhalten, dass der Nutzer eine VPN Verbindung zu seinem Unternehmen herstellen möchte. Daher stellt der RSP eine Verbindung zum Unternehmen her. Nachdem das Unternehmen und der RSP

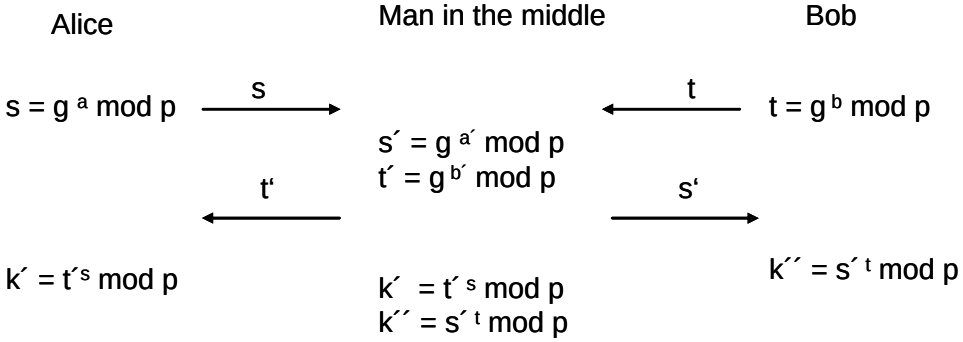
sich gegenseitig authentifiziert haben, leitet der RSP die den Nutzer betreffende Authentifizierungsinformation zum Unternehmen. Diese Authentifizierungsdaten können als Nachweis dienen, so dass der Heim-ISP und der RSP nachprüfen können, ob der kontaktierte ISP den Endnutzer wirklich authentifiziert hat.

Danach erzeugt der RSP die VPN Schlüssel für das Unternehmen und den Nutzer. Der RSP kann einen Sitzungsschlüssel für beide Richtungen oder einen Sitzungsschlüssel jeweils für jede Richtung erzeugen. Schließlich verteilt er die Sitzungsschlüssel an beide Kommunikationspartner, das Unternehmen und den Mitarbeiter bzw. Endnutzer. Die Schlüssel sollten auf eine sichere Weise verteilt werden. Der RSP sollte die Schlüssel signieren und danach mit dem öffentlichen Schlüssel des Nutzers und des Unternehmens verschlüsseln. Alternativ könnten Unternehmen und Mitarbeiter prinzipiell auch mit dem in Abbildung 74 dargestellten immer noch häufig eingesetzten Diffie-Hellmann Schlüsselaustausch ohne Hilfe des RSP Schlüssel austauschen. Dies birgt jedoch die Gefahr eines „Man-in-the-middle“ Angriffs, wie er in Abbildung 75 dargestellt ist.



Alice und Bob haben nun den gemeinsamen Schlüssel $k = g^{(a\,b)} \bmod p$

Abbildung 74 Diffie-Hellmann (DH) Schlüsselaustausch



Der „Man in the middle“ hat nun mit Alice den gemeinsamen Schlüssel $k' = g^{(a\,b')} \bmod p$ und mit Bob den gemeinsamen Schlüssel $k'' = g^{(a'b)} \bmod p$. Er kann nun problemlos jeweils die Nachrichten von Alice und Bob empfangen, entschlüsseln und mit dem entsprechenden anderen Schlüssel wieder verschlüsseln und weiterschicken, so dass Alice und Bob seine Anwesenheit gar nicht bemerken.

Abbildung 75 Man-in-the-middle Angriff auf DH

Der Nutzer muss über ein von der Unternehmens-CA ausgestelltes Zertifikat verfügen. Da nicht vorausgesetzt werden kann, dass der kontaktierte ISP in der Lage ist, einen Validierungspfad zu einem seiner Vertrauensanker zu finden, bedeutet dies, dass man nicht voraussetzen kann, dass der kontaktierte ISP das Zertifikat des Nutzers akzeptiert, auch wenn dieses korrekt ausgestellt und gültig ist. Daher impliziert die Möglichkeit des Unternehmens, Zertifikate auszustellen, die Möglichkeit der Delegation der Verifikation dieser Zertifikate.

Die Lösung für diesen Fall ist der in Abschnitt 5.1.1.3 dargestellten Lösung ähnlich. Sie ist in Abbildung 76 schematisch dargestellt. Die vom Nutzer gesendete Nachricht ist jedoch anders, als bei der in Abschnitt 5.1.1.3 präsentierten Lösung. Das Zertifikat des Nutzers $cert_U$ ist hier in der ersten vom Nutzer gesendeten Nachricht enthalten. Der Grund für diese Modifikation wird klar, wenn man den Fall der Delegierung genauer betrachtet.

Da der Fokus hier darauf liegt, möglicherweise hoch sensitive Informationen sicher zwischen Unternehmen und Nutzer auszutauschen, beschränke ich meine Überlegungen im folgenden auf den Fall, in welchem der kontaktierte ISP als nicht vertrauenswürdig angesehen wird. Hier entsteht wieder das Problem der Zuordnung der Adresse des mobilen Endgerätes zur Identität des Nutzers. Diese Zuordnung muss dem RSP bekannt sein, damit er diese Information zum Unternehmen weiterleiten kann.

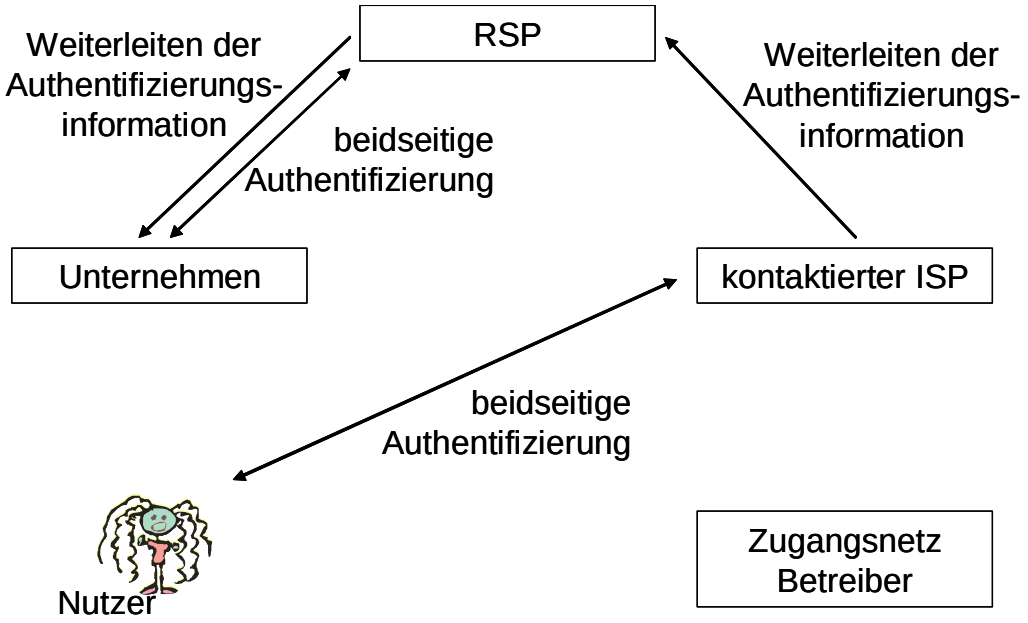


Abbildung 76: Authentifikation im Roaming VPN Model

In dieser Lösung ist der Nutzer involviert in die Erzeugung der verifizierbaren Adressinformation. Im Folgenden nutzen wir dieselbe Notation wie schon in Abschnitt 5.1.1. Der Nutzer *U* initiiert die Authentifizierung durch Senden einer „hello“ Nachricht,

welche die Identität des Nutzers *U* enthält, einen Zufallswert r_U generiert vom Nutzer *U*, und das Zertifikat des Nutzers $cert_U$. Auf den Empfang der „hello“ Nachricht hin erzeugt *B* eine $cert_B,r_B,U,t,sig_B(r_U,r_B,u,t)$ Nachricht . Dies setzt voraus, dass *U* die Adresse ad_U für sein mobiles Endgerät von *B* erhalten hat. Dann antwortet *U* mit der Nachricht $B,t,ad_U,sig_U(r_U,r_B,B,t,ad_U)$. Der Nutzer *U* sollte die Nachricht nur signieren, wenn der von *B* erzeugte Zeitwert akzeptabel ist und die angegebene Adresse ad_U gültig ist; wenn nicht sollte *U* das Protokoll abbrechen. Das Protokoll ist in Abbildung 77 dargestellt. Es handelt sich dabei um eine Variante des in Abbildung 54 dargestellten Protokolls. Der Vorteil der Variante hier besteht darin, dass der ISP B schon nach der ersten vom Nutzer geschickten Nachricht das Nutzerzertifikat auf Gültigkeit überprüfen kann und bei negativem Ergebnis gegebenenfalls die dritte Nachricht des Protokolls gar nicht mehr erzeugen sowie schicken muss. Der Vorteil der Variante aus Abbildung 54 gegenüber der hier dargestellten ist, dass bei der Weiterleitung der Authentifikationsinformation, wie in Abbildung 55 gezeigt, die letzte Nachricht des Protokolls weitergeleitet werden kann, während bei der Variante hier die Nachricht zur Weiterleitung zusammengesetzt werden muss aus den im Protokoll erhaltenen Informationen.

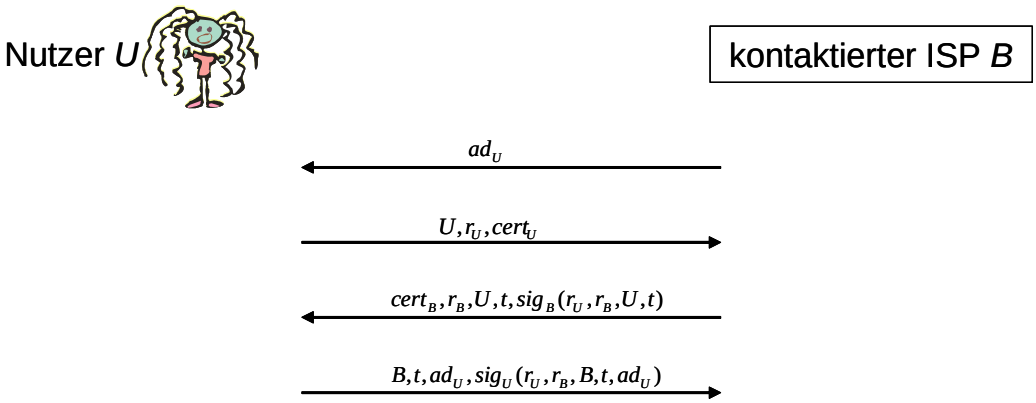


Abbildung 77: Protokoll zur Authentifikation bei nicht vertrauenswürdigem ISP B

Zwecks Authentifizierung und Überprüfung der Adresse schickt *B* die erhaltene Information innerhalb eines Authentifizierungsprotokolls zum RSP *R*, welcher die Information dann weiterleiten kann zum Unternehmen *E*. Diese Information kann dann vom RSP *R* oder Unternehmen *E* genutzt werden, um zu verifizieren, dass *B* wirklich den Nutzer *U* authentifiziert hat und dass die Adresse ad_U von *U* 's Endgerät korrekt ist. Dies ist in Abbildung 78 dargestellt.

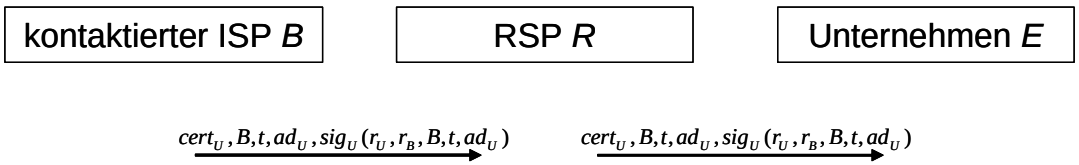


Abbildung 78: Weiterleitung der Nutzer Authentifikationsinformation gepaart mit Adressenzuordnung.

Für die beidseitige Authentifizierung zwischen Unternehmen und RSP kann eine einfachere Lösung angewendet werden. Um dieses Problem zu lösen kann die in Abbildung 52 dargestellte Lösung eingesetzt werden. Der Zeitwert ist nicht obligatorisch bei dieser Lösung. Die Parteien sollten so ausgetauscht werden, dass der RSP das Protokoll initiiert. Er entspricht also dem Nutzer U in Abbildung 52.

5.2.1.1 Delegation der Authentifikation zum RSP

Die in diesem Abschnitt dargestellte Lösung ist relevant, wenn der kontaktierte ISP nicht in der Lage ist, den Nutzer zu authentifizieren. Die Authentifizierung des Nutzers wird dann vom kontaktierten ISP zum RSP delegiert. Diese Delegation wird aus den zwei folgenden Gründen notwendig: Zum einen könnte der kontaktierte ISP nicht die notwendigen Algorithmen bzw. Mechanismen zu Verfügung haben, und zum anderen könnte es sein, dass der kontaktierte ISP nicht das Zertifikate des Nutzers akzeptiert, da er nicht in der Lage ist, einen Validierungspfad von der Unternehmens-CA zu einem seiner Vertrauensanker herzustellen. Bezüglich der Authentifizierung entspricht die Lösung der in Abschnitt 5.1.1.3.1 bereits beschriebenen Lösung bis auf die Tatsache, dass hier zusätzlich eine beidseitige Authentifizierung zwischen Unternehmen und RSP vorgenommen wird. Dies wird in Abbildung 79 dargestellt.

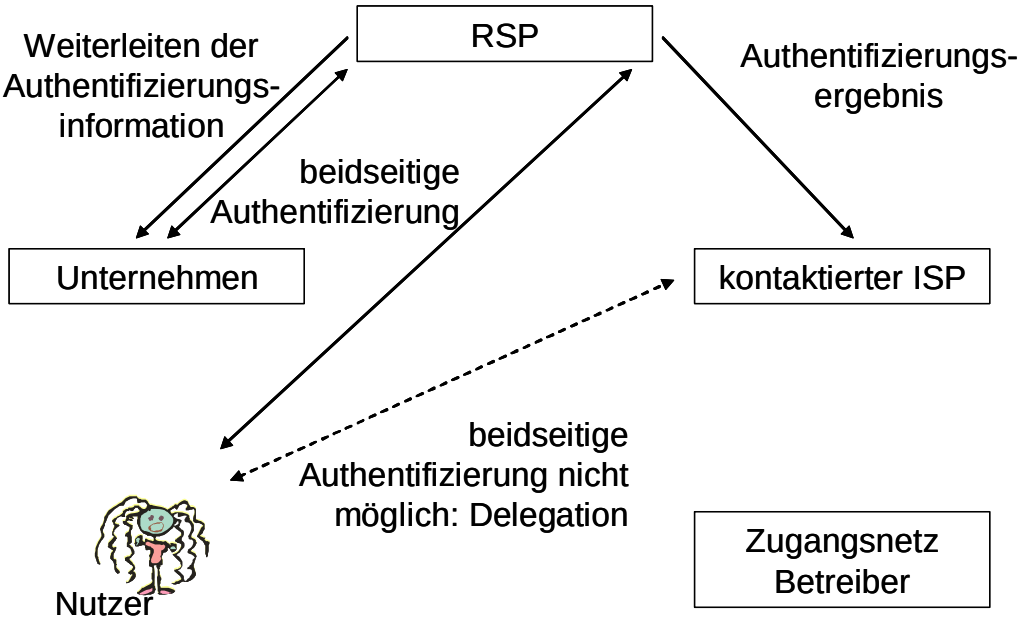


Abbildung 79: Delegation der Nutzer Authentifikation zum RSP

Was die Erzeugung und Verteilung der VPN-Schlüssel angeht, ist die Lösung mit Delegation identisch zu der Lösung ohne Delegation aus Abschnitt 5.2.1. Die Delegation hat keinerlei Einfluss auf die Schlüsselverteilung.

Die vom Nutzer gesendeten Nachrichten sind hier im Vergleich zu Anschnitt 5.1 modifiziert. Der Grund dafür liegt darin, dass diese Modifizierungen im Verhältnis stehen zu den Anforderungen im Falle des kontaktierten ISP akzeptiert keine von der Unternehmens CA ausgestellten Zertifikate. In der in Abschnitt 5.1 beschriebenen Lösung wird das Zertifikat des Nutzers in der zweiten vom Nutzer geschickten Nachricht gesendet. Der kontaktierte ISP kann die Delegation erst initiieren, wenn er die zweite Nachricht vom Nutzer erhalten hat. Wenn der Nutzer sein Zertifikat schon in der ersten Nachricht sendet kann der ISP die Delegation der Authentifizierung vergleichsweise früher einleiten für den Fall, dass er das Zertifikat des Nutzers nicht selbst akzeptiert. Das Protokoll der zum RSP delegierten Authentifikation ist in Abbildung 80 dargestellt.

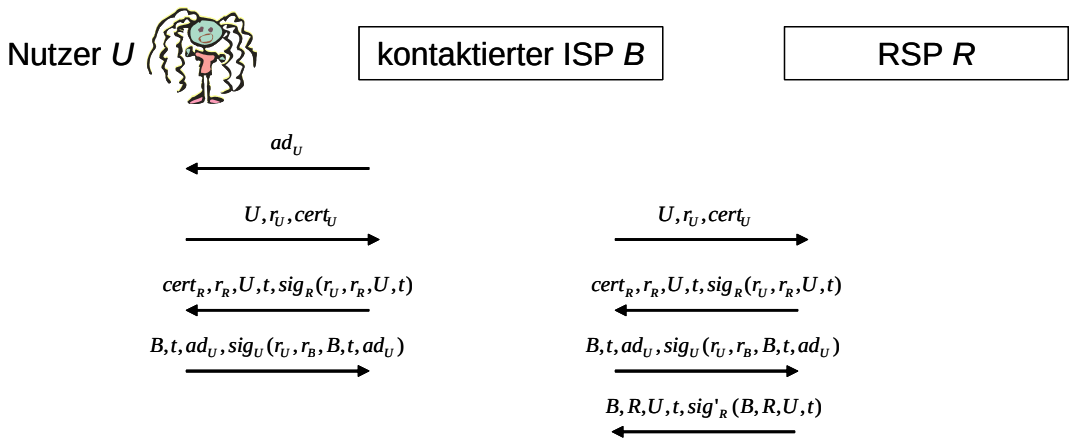


Abbildung 80: Delegierte Nutzer Authentifikation durch den RSP

5.2.2 Getrennte Authentifizierung durch Zugangsnetzbetreiber und kontaktierten ISP

Der Lösung dieses Abschnitts liegt das Szenario eines umherreisenden Mitarbeiters, der ein VPN zu seinem Unternehmen aufbauen möchte, zugrunde. Im Unterschied zu Abschnitt 5.2.1, benötigen hier sowohl der Zugangsnetzbetreiber als auch der kontaktierte ISP eine Authentifizierung des Nutzers. Dies ist schematisch in Abbildung 81 dargestellt.

Die hier betrachtete Lösung ist relevant in einem Szenario, in welchem der Nutzer als Unternehmensmitarbeiter oder sein Unternehmen einen Vertrag mit dem Zugangsnetzbetreiber haben und daher der Zugangsnetzbetreiber den Nutzer authentifizieren muss. In einem solchen Szenario kann man davon ausgehen, dass der Zugangsnetzbetreiber das von der Unternehmens-CA ausgestellte Zertifikat des Nutzers erkennt. Andernfalls muss eine andere Art der Authentifizierung zwischen Zugangsnetzbetreiber und Nutzer angewandt werden.

In einem solchen Szenario ist die Nutzerauthentifizierung, welche vom Zugangsnetzbetreiber durchgeführt wird, außerhalb des hier betrachteten Geschäftsmodells „Roaming VPN Access“. Die Lösung für das Authentifizierungsprotokoll zwischen Nutzer und kontaktiertem ISP sowie RSP und Unternehmen wird von dem zusätzlichen Authentifizierungsprotokoll zwischen Nutzer Zugangsnetzbetreiber nicht beeinflusst. Das bedeutet, dass für den Teil des Authentifizierungsprotokolls, in den der kontaktierte ISP, der RSP und das Unternehmen involviert sind, die in Abschnitt 5.2.1 dargestellte Lösung angewandt werden kann.

Die Kosten der Authentifizierung sind für die in diesem Abschnitt dargestellte Lösung alles in allem höher als die Kosten für die in Abschnitt 5.2.1 vorgeschlagene Lösung, da zwei Authentifizierungen durchgeführt werden. Die die Schlüsselverteilung zur Etablierung der VPN Verbindung zwischen Untenehmen und Mitarbeiter betreffenden Fragen bleiben davon unberührt.

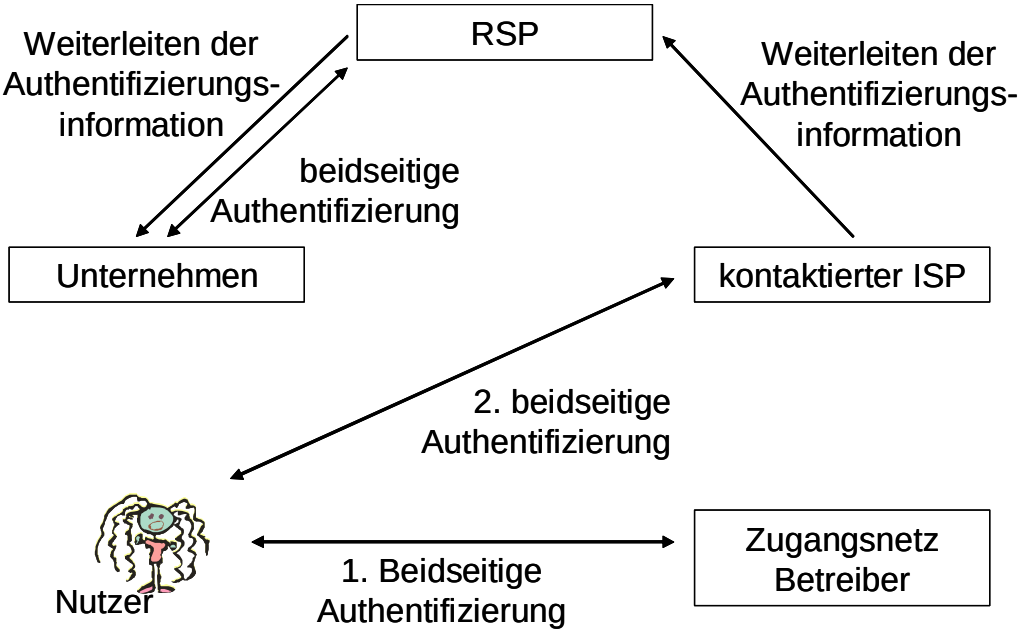


Abbildung 81: Getrennte Authentifikation durch AN Betreiber und kontaktierten ISP

5.2.2.1 Delegation der Authentifizierung vom kontaktierten ISP zum RSP

Hier wird der Fall der Delegation der Authentifizierung vom kontaktierten ISP zum RSP diskutiert. Die Lösung dieses Falles ist der in Abschnitt 5.2.1.1 ähnlich. Der Unterschied besteht darin, dass es eine zusätzliche Authentifizierung zwischen Zugangsnetzbetreiber und Nutzer gibt. Dies ist schematisch in Abbildung 82 dargestellt. Wie schon in Abschnitt 5.2.2 erwähnt, ist die zusätzliche Authentifizierung zwischen Zugangsnetzbetreiber und Nutzer außerhalb der hier betrachteten Geschäftsmodelle.

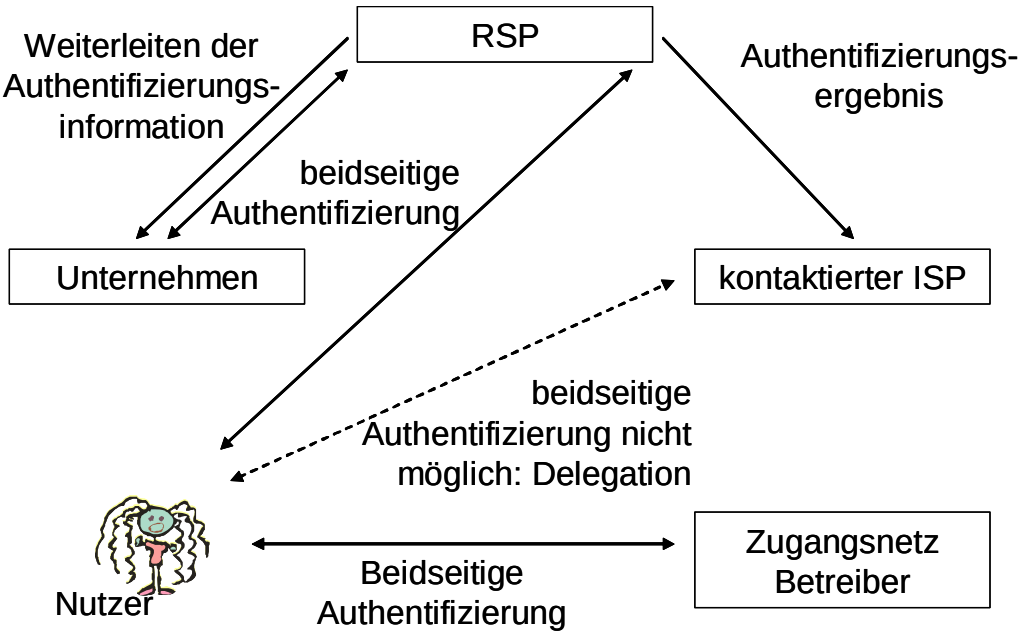


Abbildung 82: Getrennte Authentifikation durch Zugangsnetzbetreiber und kontaktierten ISP mit Delegation

5.2.3 Authentifikation durch den Betreiber des Zugangsnetzes und Weiterleitung des Authentifikationsergebnisses zum kontaktierten ISP und RSP

In Abschnitt 5.2.2 wird der Fall einer zusätzlichen Authentifizierung zwischen kontaktiertem ISP und Nutzer betrachtet. In diesem Abschnitt hier wird beschrieben, wie der zusätzliche Aufwand zwischen kontaktiertem ISP und Nutzer reduziert werden kann. Der hier beschriebene Fall wird in Abbildung 83 dargestellt.

Wenn keine Delegation, sei es zur Authentifikation oder zur Verifikation benötigt wird, authentifizieren sich der Nutzer und der Zugangsnetzbetreiber gegenseitig. Das Ergebnis wird dann über den kontaktierten ISP und den RSP zum Unternehmen weitergeleitet. Dieser Ansatz ermöglicht es, den involvierten Parteien die Identität des Nutzers mit möglichst geringer Nutzer-Interaktion zu überprüfen. Bevor der kontaktierte ISP und der RSP die Authentifizierungsinformation weiterleiten, überprüfen sie, ob die erhaltene Information korrekt ist. Das Unternehmen benötigt hier wiederum irgendwelche Zuordnungsinformation, um die temporär dem Nutzer gegebene Adresse seiner Identität zuordnen zu können, wie schon in Abschnitt 5.2.1 erläutert.

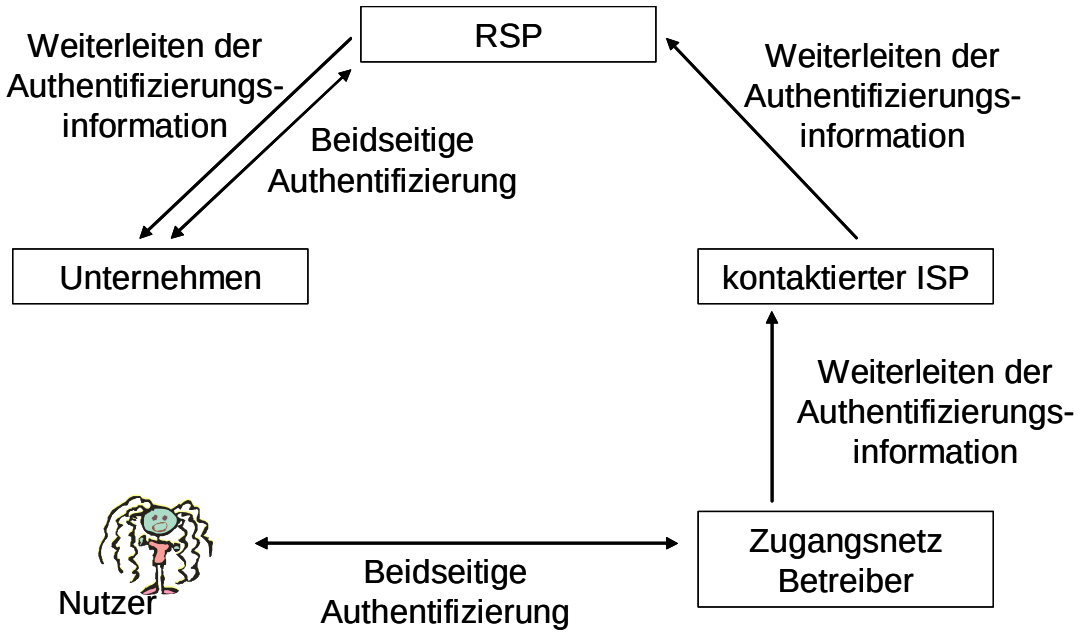


Abbildung 83: Nutzer Authentifikation und Weiterleitung der Authentifikationsinformation zum kontaktierten ISP, RSP, und Unternehmen

Die Authentifizierungslösung für diesen Fall ist in Abbildung 84 dargestellt. Wir setzen voraus, dass der Zugangsnetzbetreiber *W* den Mitarbeiter mit der Adressinformation versorgt hat bevor der Mitarbeiter seine „credentials" sendet. Möglicherweise ist der Zugangsnetzbetreiber schon vorher mit der Adressinformation vom kontaktierten ISP versorgt worden.

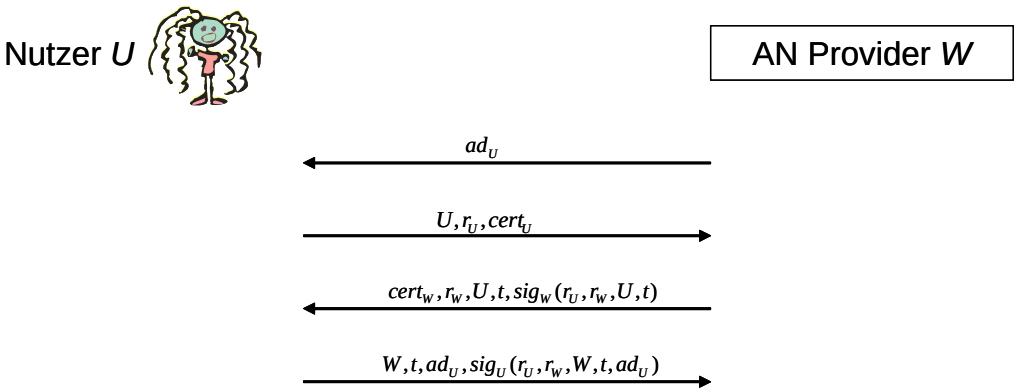


Abbildung 84: Protokoll zur beidseitigen Authentifikation zwischen Nutzer und Betreiber des Zugangsnetzes

Im nächsten Schritt leitet der Zugangsnetzbetreiber die Authentifizierungsinformation, die er vom Nutzer erhalten hat zum kontaktierten ISP weiter. Von dort wird dann die Authentifizierungsinformation zum RSP weitergeleitet um schließlich beim Unternehmen anzukommen. Dies wird in Abbildung 85 dargestellt.

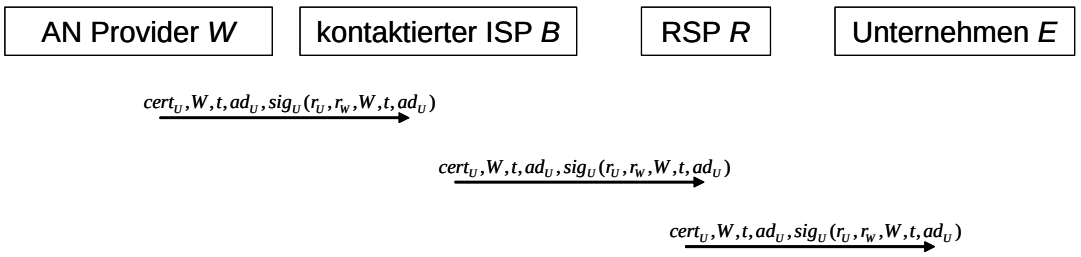


Abbildung 85: Weiterleiten der Authentifikationsinformation nach Authentifikation durch den Zugangsnetz Bereitsteller

Abhängig von den Vertrauensbeziehungen zwischen den Parteien entlang der Weiterleitungskette können einige weitere Mechanismen benötigt und eingesetzt werden, um die weitergeleitete Information zu schützen. Solche Schutzmechanismen sind vor allem für Abrechnungszwecke sinnvoll, welche hier nicht weiter betrachtet werden.

Der hier abgebildete Nachrichtenfluss kommt nur in dem Fall zur Anwendung, dass der Zugangsnetzbetreiber in der Lage ist, ohne Delegation das benötigte Authentifizierungsprotokoll abzuwickeln und einen Validierungspfad von einem seiner Vertrauensanker zur Unternehmens CA, welche das Zertifikat des Nutzers ausgestellt hat, herzustellen, um die erhaltene Authentifizierungsinformation zu verifizieren.

5.2.3.1 Delegation der Authentifikation zum RSP

Im Folgenden werden hier unterschiedliche Szenarios inklusive Delegation betrachtet. Dabei werden zwei Fälle der Delegation unterschieden:

1. Eine Partei, welche die Absicht hat, eine Authentifizierung des Nutzers durchzuführen, ist dazu nicht in der Lage. Dies könnte der Fall sein, wenn sie nicht fähig ist, die benötigten Algorithmen auszuführen oder wenn sie keinen Pfad von einem ihrer Vertrauensanker zur Unternehmens CA, welche das Zertifikat des Nutzers ausgestellt hat, zu finden.
2. Eine Partei, welche mit der Authentifizierungsinformation versorgt wird, ist nicht in der Lage diese Information zu überprüfen. Daher delegiert diese Partei die Verifikation zu einer anderen Partei. Die Gründe dafür, nicht in der Lage zu sein die weitergeleitete Authentifizierungsinformation zu überprüfen, können dieselben sein wie im ersten Fall.

Ein Beispiel, welches den ersten Fall betrifft ist gegeben, wenn der Zugangsnetzbetreiber nicht in der Lage ist, den Nutzer zu authentifizieren. Der Zugangsnetzbetreiber kann dann die Authentifizierungsarbeit zum kontaktierten ISP delegieren, welcher ihn dann mit dem Ergebnis der Authentifzierung versorgt. Des Weiteren leitet der kontaktierte ISP die

Authentifizierungsinformation zum RSP wie schon in Abschnitt 5.2.3 dargestellt weiter. Von dort aus gelangt dann die Information zum Unternehmen. Dies ist in Abbildung 86 dargestellt.

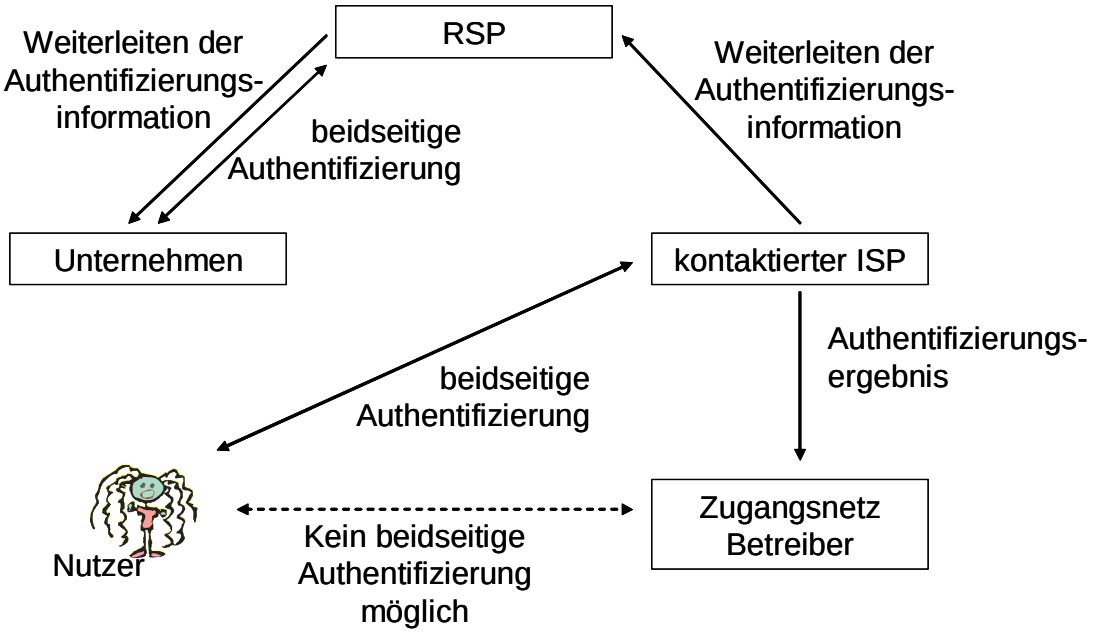


Abbildung 86: Delegation der Nutzer Authentifikation zum kontaktierten ISP

In Abbildung 87 wird der Nachrichtenfluss des Authentifizierungsprotokolls im Falle der Delegation und in Abbildung 88 die Weiterleitung der Authentifizierungsinformation skizziert. Aus demselben Grund wie schon in Abschnitt 5.2.1, wird hier das Zertifikat des Nutzers bereits in seiner ersten Nachricht an den ISP geschickt und dient als ein *a posteriori* Rechtfertigung für die Lösung aus Abschnitt 5.2.3.

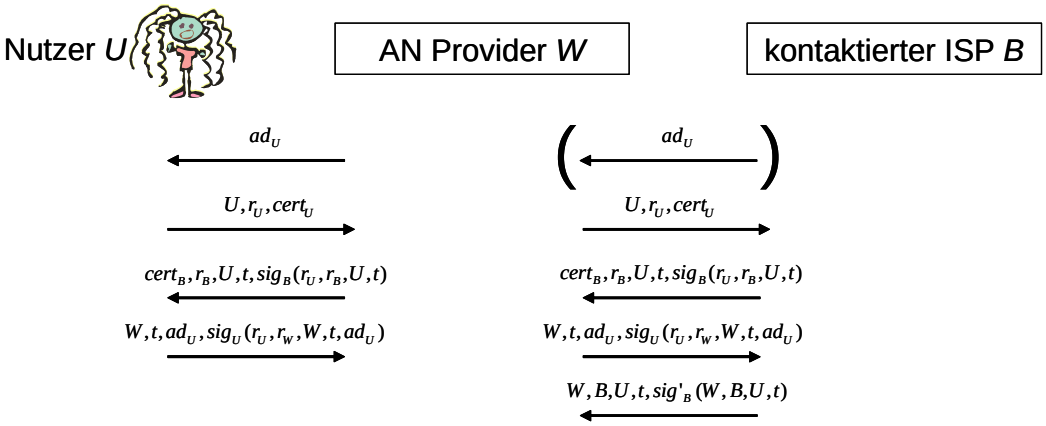


Abbildung 87: Delegation der Nutzer Authentifikation zum kontaktierten ISP

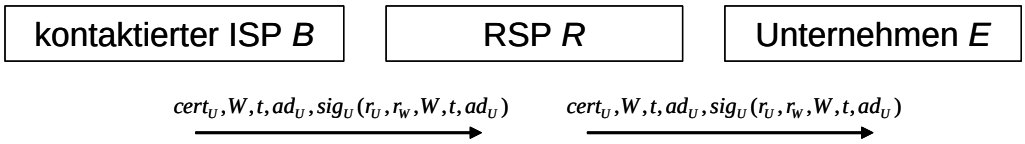


Abbildung 88: Weiterleitung der Authentifikationsinformation im Fall der Delegation

Bisher wird hier die einfache Delegation betrachtet. Es kann auch den Fall geben, dass der kontaktierte ISP nicht in der Lage ist, den an ihn gerichteten Authentifizierungsrequest zu beantworten, da er die an ihn delegierte Authentifizierung nicht durchführen kann. In diesem Fall kann er die an ihn delegierte Authentifizierung noch mal weiterdelegieren zum RSP. Ich setze hier voraus, dass der RSP immer in der Lage ist die Authentifikation durchzuführen. Danach sendet der RSP as Ergebnis zum kontaktierten ISP und der kontaktierte ISP leitet es weiter zum Zugangsnetzbetreiber. Die Lösung für diese weiterdelegierte Authentifizierung ist in den folgenden Abbildung 89 dargestellt. Das Authentifikationsprotokoll ist in Abbildung 90 und die Weiterleitung der Authentifizierungsinformation in Abbildung 91 dargestellt.

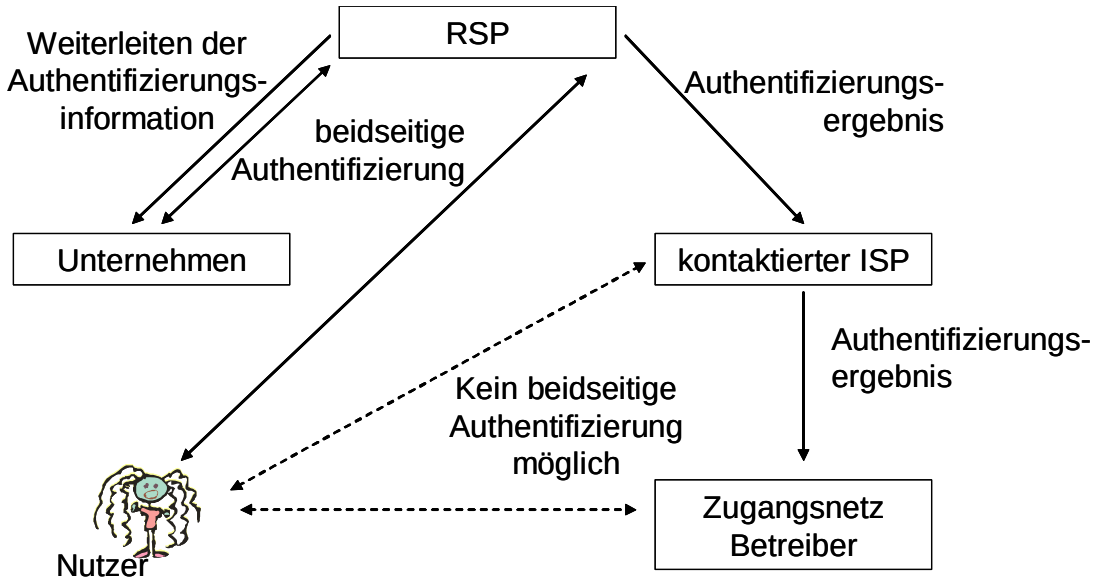


Abbildung 89: Authentifikation durch RSP nach Re-Delegation

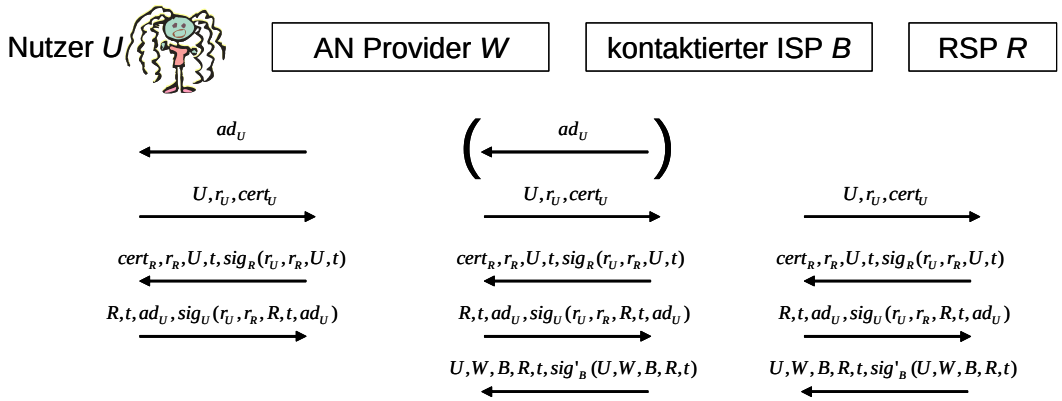


Abbildung 90: Authentifikationsprotokoll im Falle der Re-Delegation

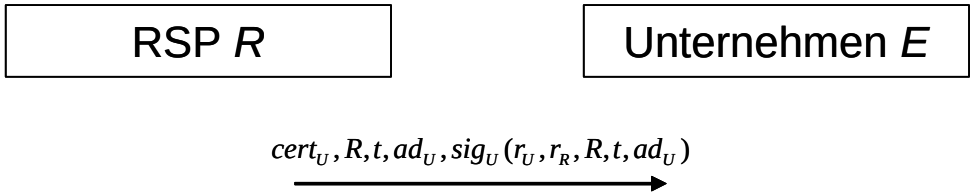


Abbildung 91: Weiterleitung der Authentifikationsinformation nach Re-Delegation

Wie schon zuvor erwähnt, behandelt der zweite Fall der Delegation das Szenario einer Partei, die nicht in der Lage ist, die weitergeleitete Authentifizierungsinformation zu überprüfen. In diesem Szenario ist es der kontaktierte ISP, der nicht in der Lage ist, die vom Zugangsnetzbetreiber bereitgestellte Authentifizierungsinformation zu überprüfen. In folgender Abbildung 92 ist eine Lösung, welche sich in diesem Szenario anwenden lässt, dargestellt.

Das Authentifikationsprotokoll ist in Abbildung 93 und die Weiterleitung der Authentifizierungsinformation in Abbildung 94 dargestellt.

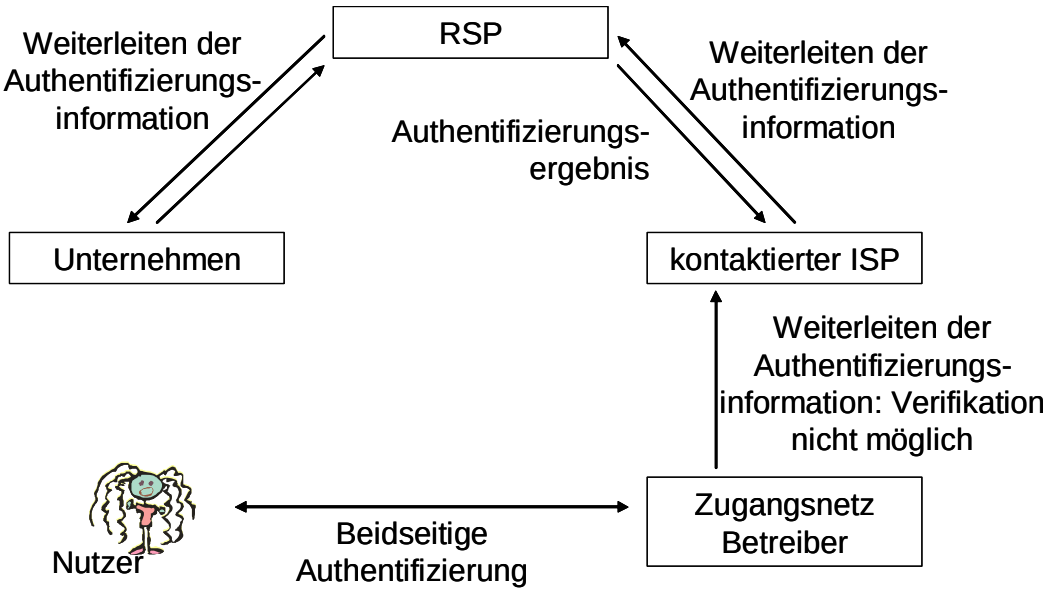


Abbildung 92: Delegation der Verifikation durch kontaktierten ISP

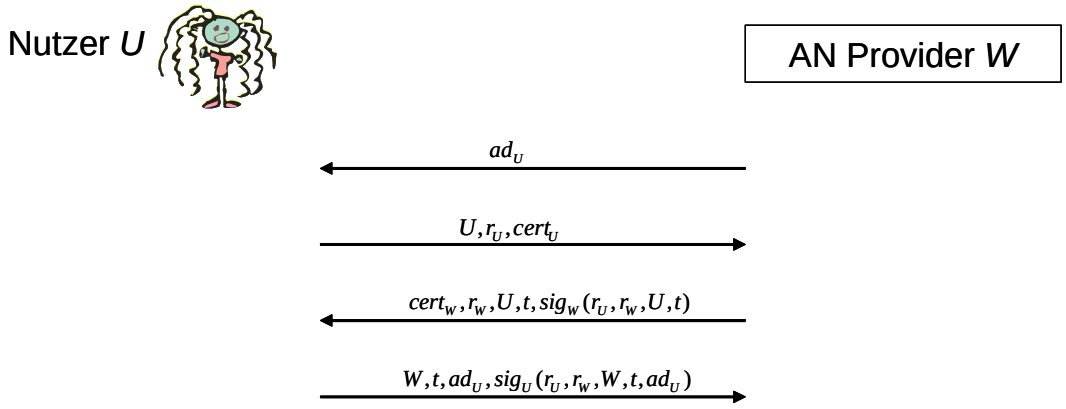


Abbildung 93: Nutzer Authentifikation durch Zugangsnetz Betreiber

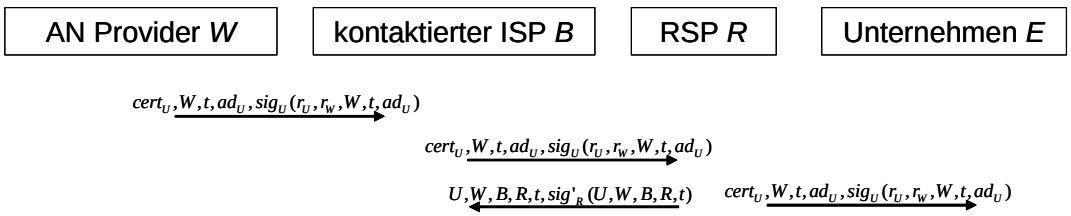


Abbildung 94: Weiterleitung der Authentifikationsinformation im Falle der Verifikations-Delegation

5.3 Seamless Roaming VPN Modell mit internem HomeAgent

Das Seamless Roaming Modell mit VPN Zugang beruht auf dem Einsatz von Mobile IP. Dabei sollten sowohl Mobile IPv4 als auch Mobile IPv6 berücksichtigt werden. Die abstrakte Lösung, welche in diesem Kapitel beschrieben wird, erfordert keine Differenzierung zwischen den beiden IP Versionen.

Im hier bedachten Szenario betreibt bzw. verwaltet das Unternehmen die HomeAgents für seine Mitarbeiter selbst. Ein Unternehmensangestellter als Nutzer stellt eine Verbindung zum Internet über einen möglichen fremden ihm unbekannten ISP her, um eine übergangslose Verbindung zu haben. Über diese Verbindung errichten das Unternehmen und der Angestellte einen VPN Tunnel zueinander. Die Erzeugung des VPN Tunnels wird vom RSP unterstützt, welcher Sitzungs-Schüssel erzeugt und verteilt.

Im werden verschiedene Ansätze präsentiert, die verwendet werden können bevor eine VPN Verbindung etabliert ist. Diese Ansätze unterscheiden sich bezüglich ihrer Kosten für den Authentifizierungsworkload und der Benutzerfreundlichkeit. Sie können auf Basis existierender Standards realisiert werden, was einen positiven Einfluss auf die Implementierungskosten hat. Es lassen sich hier die folgenden Lösungsansätze unterscheiden:

- Nutzerauthentifizierung durch den kontaktierten ISP und Standard MIP Nutzerauthentifizierung des HA
    - und ggf. Nutzerauthentifizierung mit Delegation
- Kombinierte Nutzerauthentifizierung durch den kontaktierten ISP und den HA auf Basis modifizierter MIP Registration
    - und ggf. Nutzerauthentifizierung mit Delegation

Auch wenn der Fokus hier auf dem Einsatz des MIP Protokolls liegt, wird die Lösung im Folgenden auf einer abstrakten Ebene beschrieben. Im Rahmen des MIP Registrations-Protokolls werden im Folgenden nur die für die Authentifikation relevanten Teile betrachtet. Des Weiteren wird die unterhalb der IP-Schicht liegende Authentifikation des Zugangsnetzes nicht betrachtet.

5.3.1 Nutzerauthentifikation durch den kontaktierten ISP mit Standard MIP Nutzer Registration durch den HA

In diesem Abschnitt betrachte ich Ansätze, welche Standard Methoden von MIP so weit wie möglich verwenden. Dies hat den Vorteil, dass Veränderungen des MIP Standard Protokolls nicht notwendig sind. Andererseits ist die so entstehende Lösung weniger effizient.

In der Lösung hier, wird zuerst ein Protokoll zur Authentifikation des Nutzers abgewickelt und danach das MIP Registrations-Protokoll angewandt. Das initiale

Protokoll zur Nutzerauthentifikation wird zwischen dem Nutzer und dem kontaktierten ISP abgewickelt. Dann wird die Authentifizierungsinformation weitergeleitet vom kontaktierten ISP zum RSP und dann vom RSP zum Unternehmen, wo sich der HA befindet. Die weitergeleitete Information erlaubt dem RSP und dem Unternehmen zu überprüfen, dass der kontaktierte ISP wirklich den Nutzer überprüft hat. Nach diesem Authentifizierungsprozess wird die MIP Registrations-Prozedur initiiert. Dies erlaubt den Standardprozess der MIP Registrierung durchzuführen, da die Authentifizierungsanforderungen schon erfüllt sind, wenn das MIP Registrations-Protokoll initiiert ist. Die Authentifikationsbeziehungen sind schematisch in Abbildung 95 dargestellt.

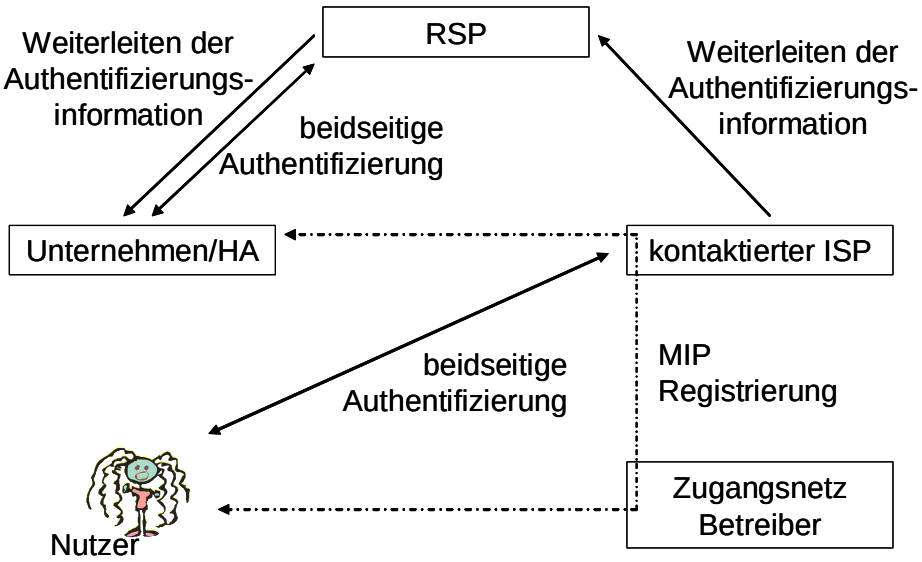


Abbildung 95: Getrennte Nutzer Authentifikation durch kontaktierten ISP und MIP Registration

Um die Lösung zu beschreiben, wird dieselbe Notation verwendet, wie im vorigen Abschnitt. Die Lösung zur beidseitigen Authentifizierung von Endnutzer und kontaktiertem ISP entspricht der Lösung aus Abschnitt 5.2.1. Sie ist in Abbildung 96 dargestellt. Der Grund für die Sendung des Zertifikates des Unternehmensmitarbeiters in der ersten vom Mitarbeiter geschickten Nachricht ist der, dass im möglichen Falle eines kontaktierten ISP, der nicht in der Lage ist, einen Pfad von einem Vertrauensanker zu der Unternehmens CA zu finden. Die Signierung der Adressinformation des Mitarbeiters erlaubt die Bereitstellung von signierter Information, welche zur Zuordnung von Adressen und Identitäten dient für das Unternehmens. Um die Identität des Nutzers zu verifizieren, überprüft der kontaktierte ISP die Gültigkeit der Signatur, welche er mit der Nachricht $cert_U, B, t, ad_U, sig_U(r_U, r_B, B, t, ad_U)$ vom Mitarbeiter erhalten hat. Nach einem positiven Verifikationsergebnis leitet der kontaktierte ISP $cert_U, B, t, ad_U, sig_U(r_U, r_B, B, t, ad_U)$ als Beweis zum RSP weiter, wie Abbildung 97 zeigt. Der RSP leitet dieselbe Nachricht dann zum Unternehmen weiter.

Während der Weiterleitung dieser Nachrichten erfolgt die MIP Registration zwischen dem Mitarbeiter und dem im Unternehmen befindlichen HA. Das Unternehmen kann die erhaltene Zuordnungsinformation benutzen, um die entsprechende Identität zur MIP Registration zu assoziieren.

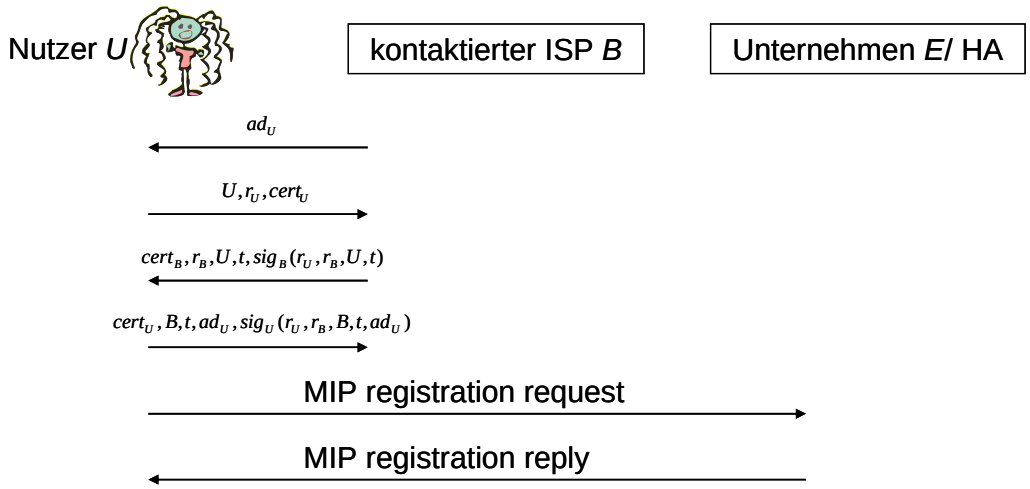


Abbildung 96: Protokoll für getrennte Nutzer Authentifikation durch kontaktierten ISP und MIP Registrierung

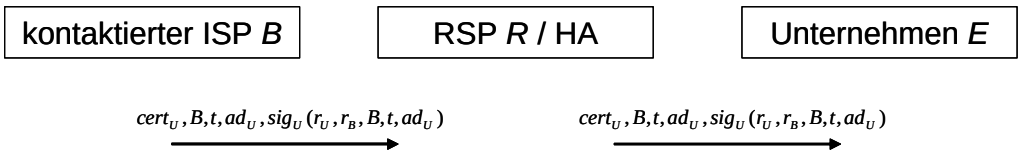


Abbildung 97: Weiterleitung der Authentifikationsinformation

Die Lösung, die bisher präsentiert wird, beinhaltet einen eigenen Vorschlag zur Authentifizierung, nutzt aber die Standardvariante der MIP Registration. Daher kann Standardsoftware für die MIP Registration angewandt werden. Der neue und proprietäre Teil der Lösung ist so entwickelt, dass die notwendige Interaktion des Nutzers bei der Authentifizierung so weit wie möglich reduziert ist. Die Identität des Nutzers kann von drei Parteien verifiziert werden – dem kontaktierter ISP, dem RSP und dem Unternehmen. Währenddessen sendet der Nutzer nur zwei Nachrichten für die Authentifizierung. Jedoch in Folge der zusätzlichen MIP Registration ist die Anzahl der insgesamt ausgetauschten Nachrichten höher.

Im Folgenden skizziere ich kurz, wie die Anzahl der insgesamt ausgetauschten Nachrichten reduziert werden kann. Dies erfordert allerdings einige neue Erweiterungen im MIP Registration Protokoll. Das Ziel ist hierbei, die vom RSP zum Unternehmen weitergeleiteten Nachrichten, welche die Information der Nutzerauthentifikation

enthalten, zu eliminieren. Dies kann erreicht werden, wenn die Signatur des Mitarbeiters zusammen mit den signierten Daten in einer Erweiterung des entsprechenden MIP Registration Protokolls eingesetzt wird. Eine derartige Erweiterung existiert bisher noch nicht und muss daher neu definiert werden. Der MIP Standard ist jedoch offen für solche Erweiterungen. Der reduzierte Nachrichtenfluss ist in Abbildung 98 gezeigt. Die Reduktion der Gesamtanzahl der Nachrichten, welche ausgetauscht werden müssen, kann also die benötigte Zeit, die ein Mitarbeiter braucht, um eine VPN Verbindung zu seinem Unternehmen zu etablieren, reduzieren.

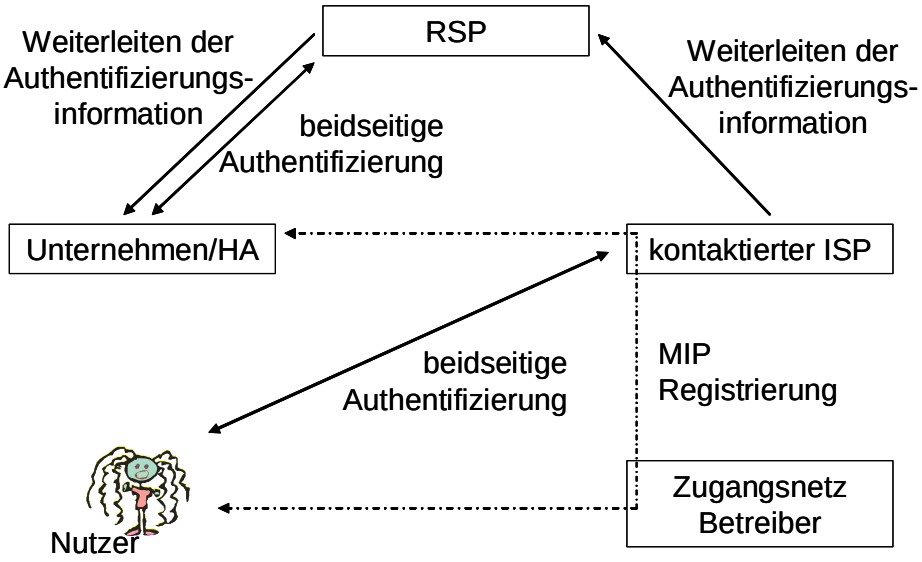


Abbildung 98: Getrennte Nutzer Authentifikation durch kontaktierten ISP und MIP Registration mit Weiterleitung der Authentifikationsinformation zum Unternehmen im MIP Registrations-Protokoll

5.3.1.1 Sicherheitsbetrachtung

Prinzipiell gilt das bereits in 5.1.1.1.1 gesagte. Um dies deutlich zu machen, wird hier stellvertretend die Sicherheit für die beiden „Seamless Roaming mit VPN Modell“ Fälle das Protokoll aus Abbildung 97 betrachtet.
Nachdem der ISP dem Nutzer eine Adresse zugeteilt hat, schickt der Nutzer seine Identität und als Challenge eine Zufallszahl an den ISP. Der Nutzer erwartet nun die Antwort seines Gegenübers. Da diese nochmals dieselbe Zufallszahl und die Identität des Nutzers enthält und diese beiden Werte in eine signierte Prüfsumme eingehen, kann sofort eine Manipulation der ersten Nachricht des Nutzers bzw. eine Manipulation seitens des kontaktierten ISPs vom Nutzer erkannt werden. Die Authentifikation wäre in diesem Fall gescheitert.

Bei korrektem Ablauf der ersten beiden Protokollschritte, d.h. der ersten beiden verschickten Nachrichten, kann der Nutzer sich Dank des Zertifikates und Dank der Signatur des kontaktierten ISP B sicher sein, dass er mit dem „richtigen“ im Zertifikat angegebenen ISP kommuniziert, falls die Überprüfung der Signatur erfolgreich war und

ein positives Ergebnis vorgelegen hat, da nur der richtige ISP B diese Signatur erstellen kann, was hier vorauszusetzen ist. Wenn die Signaturüberprüfung kein postivies Ergebnis liefert oder die signierte Prüfsumme nicht korrekt ist, ist der Authentifizierungsvorgang gescheitert, da eine Manipulation vorgelegen haben muss. Der Zeitwert sorgt zusätzlich dafür, dass ein Wiedereinspielen der Nachricht zu einem Späteren Zeitpunkt erkannt wird, was auch zum scheitern des Authentifikationsvorganges führen würde.

Analog zur dritten Nachricht des Protokolls, sendet der Nutzer an den kontaktierten ISP nun eine vierte Nachricht, die u.a. den vom ISP geschickten Zufallswert und den Zeitwert enthält. Durch die signierte Prüfsumme ist sicher gestellt, dass eine Manipulation der Nachricht erkannt werden kann. Der ISP kann die Signatur des Nutzers anhand des Nutzerzertifikates überprüfen. Falls die Signatur korrekt und das Zertifikat gültig ist, kann er sich der Identität des Nutzers sicher sein. Erst nachdem der er den Nutzer authentifiziert hat, erlaubt der ISP dem Nutzer seinen MIP Registration Request zu schicken.

Ein „Man-in-the-middle“, der versucht einerseits dem Nutzer vorzutäuschen der ISP zu sein und andererseits dem ISP vorzutäuschen der Nutzer zu sein, würde zu einem solchen vorhaben die jeweiligen privaten Schlüssel der beiden benötigen, um die entsprechenden Signaturen erstellen zu können. Solange er über diese nicht verfügt, kann er sich nicht unerkannt zwischen die beiden Kommunikationspartner setzten.

Eine beidseitige Authentifizierung, welche das Protokoll zum Ziel hat, wird also gewährleistet. Die hier gemachten Sicherheitsbetrachtungen gelten auch für die folgenden Fälle.

5.3.1.2 Delegation der Authentifikation des Nutzers

Die folgende Lösung kann angewandt werden, wenn der kontaktierte ISP nicht in der Lage ist, die Nutzerauthentifizierung durchzuführen, da er z.B. keinen entsprechenden Algorithmus einsetzen kann oder keinen Validierungspfad von einem seiner Vertrauensanker zum zur Unternehmens-CA hat, um die Gültigkeit des Zertifikates des Nutzers zu verifizieren. Es handelt sich dabei um die Delegationsvariante des in Abbildung 95 dargestellten Falles. Dabei wird davon ausgegangen, dass die Standard MIP Registrierung angewandt wird.

Wenn der kontaktierte ISP den Mitarbeiter nicht authentifizieren kann, wird die Authentifizierung an den RSP delegiert. Danach stellt der RSP für den kontaktierten ISP das Ergebnis der Authentifizierung bereit. Das Authentifizierungsergebnis wird so erzeugt, dass der kontaktierte ISP in der Lage ist, dieses Ergebnis zu verifizieren. Nach der Verifikation sollte der kontaktierte ISP das Authentifizierungsergebnis für weitere Zwecke speichern, wie z.B. zur Abrechnung. Des Weiteren leitet der RSP die Authentifizierungsinformation zum Unternehmen weiter, welches diese Information ebenso überprüfen sollte.

Neben dem proprietären Teil der Authentifizierungslösung mit Delegation, wickeln der Mitarbeiter und das Unternehmen das Standard MIP Registration Protokoll ab. Die MIP Registration wird nicht beeinflusst durch die Tatsache, dass die Authentifizierung des Nutzers an den RSP delegiert wird. Das MIP Registration Protokoll für den Delegations- und die Nicht-Delegations- Variante unterscheiden sich nicht.

Das Authentifizierungsprotokoll für den Delegationsfall zeigt Abbildung 100. Der Einschluss der Nutzer-Adressinformation in die Signatur des Nutzers dient dazu, das Unternehmen mit einer zuverlässigen Zuordnungsinformation zu versorgen. Nachdem der RSP die Nachricht $cert_U, B, t, ad_U, sig_U(r_U, r_B, B, t, ad_U)$ erhalten hat, erzeugt er eine neue Nachricht, die vom kontaktierten ISP verifiziert werden kann. In Abbildung 99 ist der Fall dargestellt, in dem der kontaktierte ISP nicht in der Lage ist, die Signatur des Mitarbeiters zu überprüfen, da er nicht den notwendigen Algorithmus sig zu Verfügung hat. Der RSP verwendet dann den Algorithmus sig', der dem kontaktierten ISP angepasst ist. Der RSP stellt also für den kontaktierten ISP die Nachricht $B, R, U, t, sig'_R(B, R, U, t)$ bereit.

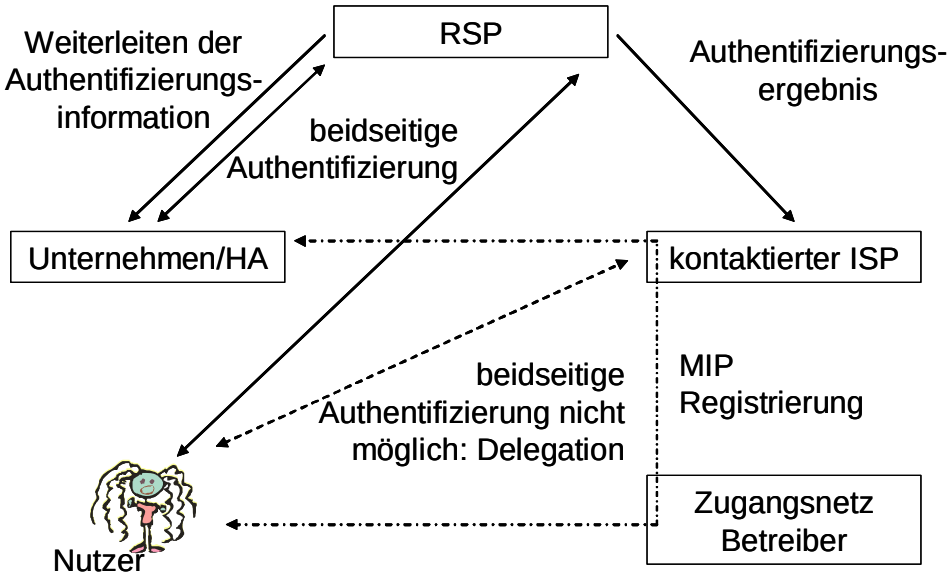


Abbildung 99: Delegation der Nutzer Authentifikation durch den kontaktierten ISP und MIP Registration

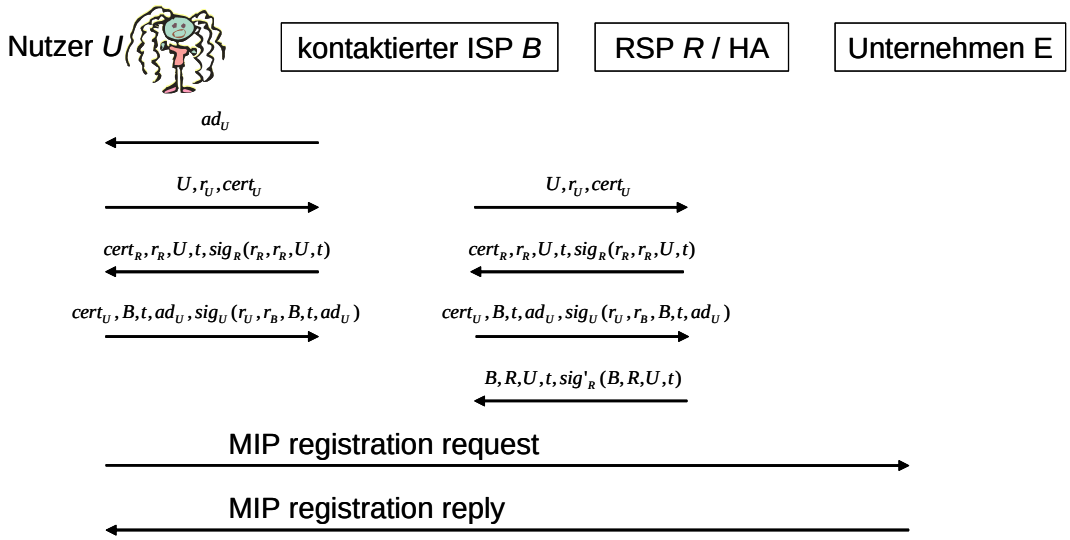


Abbildung 100: Das Protokoll im Falle der Delegation

Abbildung 101 stellt die Delegationsvariante der Lösung, bei der die Weiterleitung der Authentifizierungsinformation des Nutzers vom RSP eliminiert wird. Stattdessen wird die Information, welche vom Unternehmen benötigt wird, innerhalb des MIP Registration Protokolls in einer zusätzlichen, neu zu definierenden Extension übertragen.

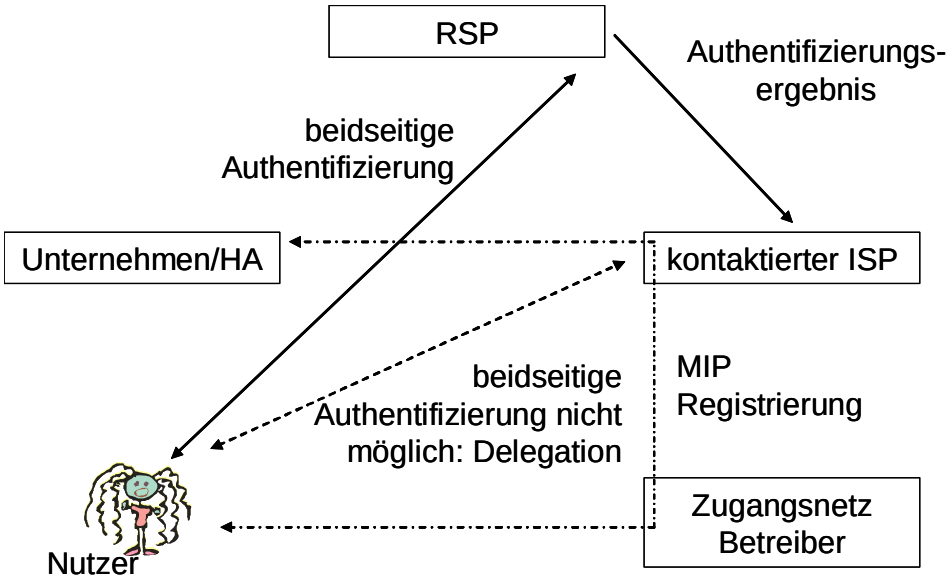


Abbildung 101: Delegation der Nutzer Authentifikation zum ISP und MIP Registration mit Weiterleitung der Authentifikationsinformation zum Unternehmen im MIP Registrations-Protokoll

5.3.1.3 Sicherheitsbetrachtung

Prinzipiell gilt das bereits in 5.1.1.1.2.1 gesagte. Um dies deutlich zu machen, wird hier stellvertretend die Sicherheit für die beiden „Seamless Roaming mit VPN Modell“ Fälle das Protokoll aus Abbildung 100 betrachtet:
Die Sicherheit der Authentifizierung wird im Delegationsfall genauso gewährleistet, wie im Falle ohne Delegation. Der kontaktierte ISP kann wie ein Angreifer in der Leitung prinzipiell zwar die über ihn geschickten Nachrichten beliebig manipulieren. Eine derartige Manipulation wird durch die im Protokoll enthaltenen signierten Hashwerte jedoch entdeckt, was einen derartigen Angriff auf einen Denial of Service Angriff, der von jedem Angreifer, der die Möglichkeit hat Pakete zu manipulieren geführt werden kann, reduziert. Die letzte Nachricht, die der RSP im Rahmen des Protokolls an den ISP schickt, sorgt dafür, dass der kontaktierte ISP vom RSP die Bestätigung der Identität des Nutzers erhält. R kann nicht plötzlich eine andere Identität von U vortäuschen, da die erklärte Identität von U dem ISP ja geschickt wurde. Der Zeitwert wurde von B selbst generiert. Daher würde eine Manipulation desselben auch von B bemerkt werden. Was die Überprüfung der Identität angeht, muss B dem RSP R vertrauen, wie in Kapitel 3 bereits ausgeführt. Der RSP kann also prinzipiell einem Nutzer, dessen Identität im Rahmen der Signatur und Zertifikatsüberprüfung nicht bestätigt werden konnte, trotzdem dem ISP B gegenüber bestätigen. Damit würde er sich allerdings selber schaden, da der ISP B ja von ihm auf Basis dieser Bestätigung später für seine Dienste Geld verlangt. Der RSP hat prinzipiell auch die Möglichkeit korrekt identifizierte Nutzer dem ISP gegenüber als nicht identifiziert zu melden. Dies käme einem „Denial of Service“ gegenüber dem Nutzer gleich, womit der RSP sich auch selber Schaden würde, da er letztlich daran verdient, dass er mit seinem Dienst den Nutzern das Roaming ermöglicht. Insofern ist ein Betrug von Seiten des RSP unwahrscheinlich. Wenn der Nutzer authentifiziert ist, erlaubt der ISP ihm erst seinen MIP Request zu schicken.

5.3.2 Kombinierte Nutzerauthentifikation durch den kontaktierten ISP und den HA auf Basis einer modifizierten MIP Registration

Die hier beschriebene Variante stellt eine effizientere Lösung dar. Die Lösung erfüllt die Anforderung der Hochgeschwindigkeitsauthentifikation besser als die in Abschnitt 5.3.1 dargelegte Lösung. Das wird dadurch erreicht, dass die Nutzerauthentifizierung in die MIP Registrierung eingebunden wird. Auf der anderen Seite benötigt ein effizienterer Ansatz eine neue Implementierung des MIP Registrations-Protokolls. Die Lösung ist in der folgenden in Abbildung 102 schematisch dargestellten.

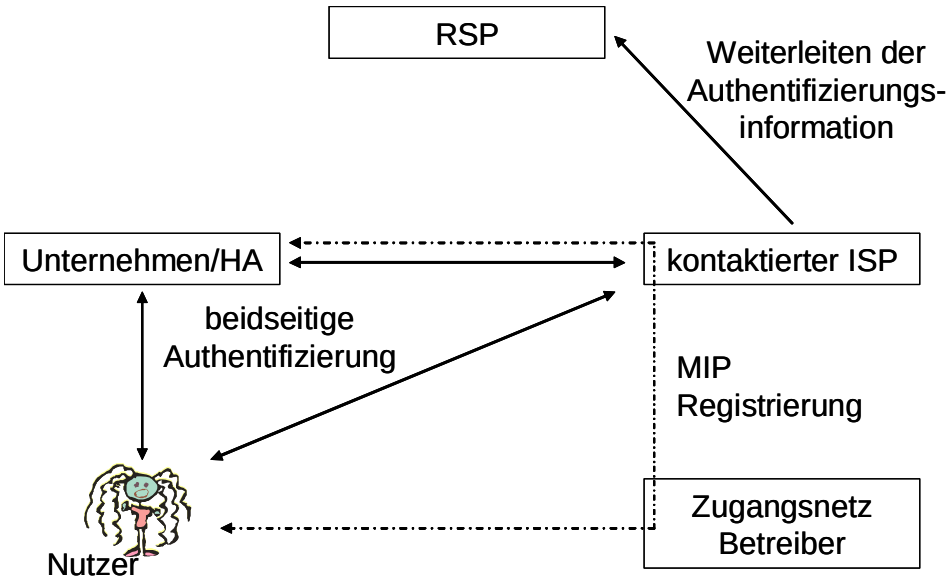


Abbildung 102: Nutzer Authentifikation durch kontaktierten ISP und durch HA auf Basis modifizierter MIP Registration

In der hier gezeigten Lösung, authentifizieren der Nutzer, der kontaktierte ISP und das Unternehmen einander. Zusätzliche Notationen müssen hierfür eingeführt werden:

- Die Adressinformationen *ad* stellt der kontaktierte ISP dem Nutzer für die Erzeugung seiner Adresse zu Verfügung. In einem MIP Szenario sind dies z. B. Informationen, die zur Erzeugung von CoAs benutzt werden.
- Die Adressinformationen *ad'* werden in MIP Szenarien verwendet, z.B. zur Darstellung von HA Adresse, Home Adresse und CoA

Das kombinierte Protokoll ist in folgender Abbildung 103 dargestellt.

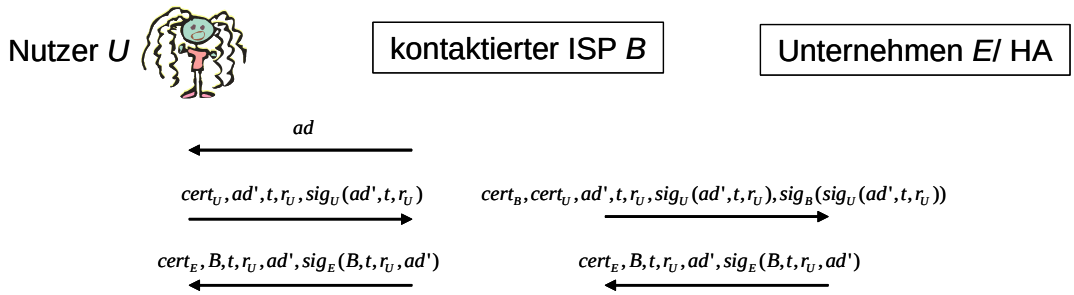


Abbildung 103: Kombiniertes Protokoll für Einbettung der Nutzer Authentifikation in die MIP Registration

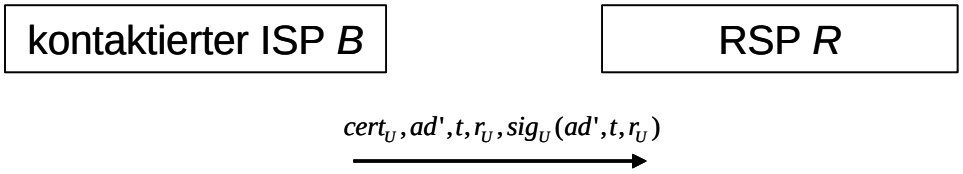


Abbildung 104: Weiterleitung der Nutzer Authentifikationsinformation to RSP

Nachdem er mit der Adresse ad des kontaktierten ISPs versorgt wird, erzeugt der Nutzer die Adressinformation ad', einen Zufallswert r_U, und den gegenwärtigen Zeitwert t. Beides r_U und t können zusammen als "Einmalwert" – nonce – angesehen werden. Dann generiert der Nutzer eine Signatur aus ad', r_U , und t. Wenn die Überprüfung der Signatur ein positives Ergebnis liefert, hat der kontaktierte ISP den Mitarbeiter authentifiziert. Im Gegensatz zu den anderen bisher in dieser Arbeit entwickelten Lösungen wird t hier auf der Nutzerseite vom Nutzer des Unternehmens erzeugt. Dies trägt den Gegebenheiten des MIP Registration Standards Rechnung. Jedoch ist es auch möglich, den kontaktierten ISP t wählen zu lassen und diesen Wert zusammen mit ad verteilen zu lassen. Aus Sicht des Nutzers entsprechen die Werte ad', r_U, und t einer Challenge im Rahmen eines Challenge-Response Protokolls.

Sobald der kontaktierte ISP die Signatur des Nutzers erhalten hat, verifiziert er sie. Wenn die Verifikation der Signatur ein positives Ergebnis ergibt, erzeugt der kontaktierte ISP eine neue Nachricht, welche zum HA im Unternehmen gesendet wird. Diese Nachricht besteht als Grundlage aus der Nachricht, welche der kontaktierten ISP vom Unternehmensmitarbeiter erhalten hat und zusätzlich einer vom kontaktierten ISP erzeugten Signatur. Diese neue Signatur wird vom Unternehmen verwendet, um den kontaktierten ISP zu authentifizieren. Das Unternehmen antwortet mit einer Nachricht, welche B, ad', r_U, und t enthält, sowie eine aus diesen Werten vom Unternehmen erzeugte Signatur, welche zum kontaktierten ISP geschickt wird. Der kontaktierte ISP verifiziert die Signatur der erhaltenen Nachricht und durch die Signierung der Daten, welche dem kontaktierten ISP schon bekannt sind, kann der kontaktierte ISP das Unternehmen authentifizieren. Der kontaktierte ISP leitet die erhaltene Nachricht weiter zum Endnutzer. Da der Nutzer als Mitarbeiter eine Nachricht mit der Signatur von dem Unternehmen, in dem er angestellt ist, erhalten hat, welche die Identität des kontaktierten ISPs bestätigt und da er als Mitarbeiter seinem Unternehmen vertraut hinsichtlich der Korrektheit dieser Angaben, geht der Nutzer davon aus, dass er wirklich mit dem entsprechenden kontaktierten ISP kommuniziert. Auf diese Weise können der Nutzer, das Unternehmen und der kontaktierte ISP einander mit Hilfe einer entsprechenden Anpassung des MIP Registration Protokolls authentifizieren.

Der kontaktierte ISP muss nun noch eine Nachricht mit den Authentifizierungsdaten zum RSP schicken, wie in Abbildung 104 dargestellt. Der RSP muss den Endnutzer authentifizieren, um die entsprechenden VPN Schlüssel zum Endnutzer und zum

Unternehmen schicken. Die Zieladresse, an die die VPN Schlüssel geschickt werden müssen, kann vom RSP aus der weitergeleiteten Nachricht extrahiert werden.

Die hier vorgeschlagene Lösung ist eine Erweiterung des MIP Registrierungs-Protokolls. Diese Lösung ist sehr effizient, aber die Standardversion des MIP Registrations-Protokolls muss erweitert werden.

5.3.2.1 Delegation der Authentifikation des Nutzers

In diesem Abschnitt wird eine Variante der Lösung des vorigen Abschnitts betrachtet. Der kontaktierte ISP ist in dem hier zugrunde gelegten Szenario nicht in der Lage, die Authentifizierung des Nutzers selbst durchzuführen. Daher delegiert er die Durchführung der Authentifizierung zum RSP.

In Abbildung 105 wird dementsprechend ein Ansatz dargestellt, in dem die Authentifizierung des Nutzers vollständig im Rahmen des MIP Registration Protokolls durchgeführt wird.

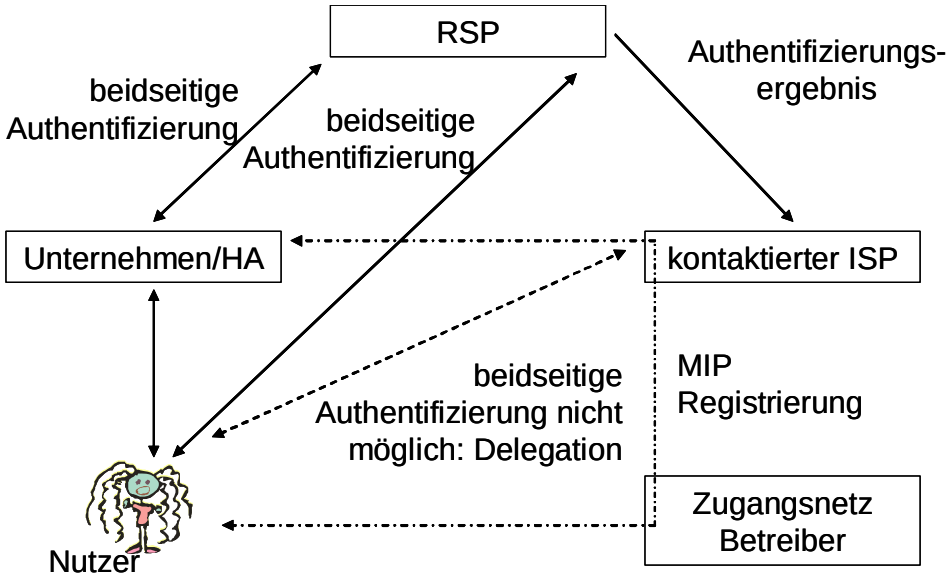


Abbildung 105: Delegierte Nutzer Authentifikation unter Benutzung modifizierter MIP Registration

Das entsprechende Authentifizierungsprotokoll zeigt die folgende Abbildung 106.

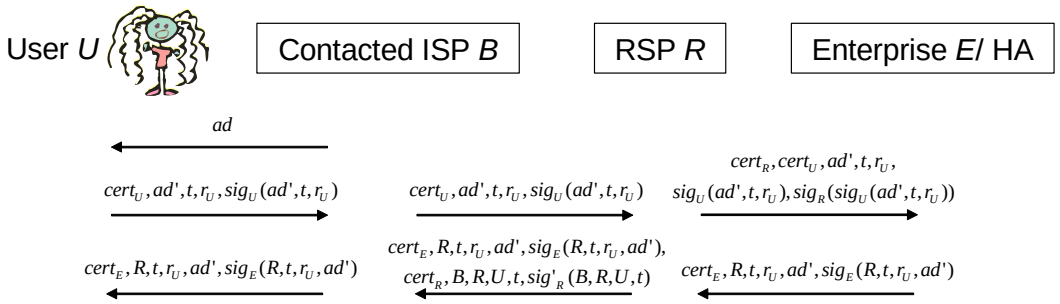


Abbildung 106: Protokoll zur delegierten Nutzerauthentifizierung mit modifizierter MIP-Registrierung

Hier wird wieder dieselbe Notation wie schon in Abschnitt 5.3.2 verwendet. Wenn der kontaktierte ISP seine Nachricht erhalten hat und nicht in der Lage ist, die Identität des Nutzers zu verifizieren, dann leitet er exakt dieselbe Nachricht, die er erhalten hat, zum RSP weiter. Der RSP signiert die Signatur des Nutzers noch mal, hängt die neue Signatur an die Nachricht an und leitet diese neue Nachricht zum Unternehmen/HA. Dann erzeugt das Unternehmen eine neue Nachricht, welche R, ad', r_U, und t enthält, signiert die neue Nachricht und schickt diese als Antwort zum RSP. Der RSP fügt eine neue Signatur an die Nachricht an. Diese Signatur ist mit einem geeigneten Mechanismus so erzeugt, dass der kontaktierte ISP sie verifizieren kann. Danach leitet der RSP die neue Nachricht zum kontaktierten ISP weiter. Der kontaktierte ISP extrahiert den für die Nutzerauthentifikation relevanten Teil und schickt die restliche Nachricht an den Nutzer.

Schließlich verfügt der RSP über die nötigen Adressen, und kann die VPN-Schlüsselpaare erzeugen und an das Unternehmen und den Nutzer als Unternehmensmitarbeiter verteilen, was den beiden ermöglicht, eine sichere VPN Verbindung zu etablieren.

Bei diesem Ansatz wird die Delegation der Nutzerauthentifikation ausschließlich innerhalb einer modifizierten Version des MIP Registration Protokolls ausgetragen. Hier im Falle der Delegation fließen die Nachrichten über eine oder mehr Parteien in jede Richtung, so wie es im MIP Registration Protokoll der Fall ist. Da es keinen höheren Interaktionsaufwand für den Nutzer bedeutet. ist dies immer noch effizient,

5.4 Seamless Roaming VPN Modell mit externem Home Agent

Beim dem “Seamless Roaming VPN Modell mit externem Home Agent” ist die Situation mit der in Abschnitt 5.3 beschriebenen Modell mit internem HA vergleichbar. Der Unterschied besteht darin, dass der HA vom RSP betrieben bzw. verwaltet wird. Dieses Szenario ist vor allem für kleine und mittlere Unternehmen, die keinen eigenen HA betreiben wollen, relevant.

Das Ziel dieses Modells ist identisch mit dem des in Abschnitt 5.3 beschriebenen Modells. Nach der Authentifizierung von sowohl Unternehmen als auch Mitarbeiter

erzeugt der RSP die VPN Schlüssel und verteilt diese, um die Etablierung einer VPN Verbindung über MIP zu ermöglichen. Wir unterscheiden hier dieselben Fälle wie im vorigen Abschnitt:

- Nutzerauthentifizierung durch den kontaktierten ISP und Standard MIP Nutzerauthentifizierung des HA
    - und ggf. Nutzerauthentifizierung mit Delegation
- Kombinierte Nutzerauthentifizierung durch den kontaktierten ISP und den HA auf Basis modifizierter MIP Registration
    - und ggf. Nutzerauthentifizierung mit Delegation

Im Allgemeinen sind die Lösungen für die Modelle hier identisch mit den Lösungen aus Abschnitt 5.3. Daher werden hier die Details nicht noch einmal erläutert. Stattdessen wird die Lösung kurz skizziert.

Es werden bezüglich des MIP Registrations-Protokolls nur die Teile betrachtet, die für die Authentifizierung relevant sind. Die Authentifizierung des Zugangsnetzes wird nicht weiter betrachtet.

5.4.1 Nutzerauthentifikation durch den kontaktierten ISP und Standard MIP Nutzer Registration durch den HA

Das Ziel dieses Abschnitts ist es, eine Lösung zu zeigen, welche die MIP Registrierung ohne Modifikationen benutzt. Der Vorteil dieses Ansatzes besteht darin, dass keine Modifikationen am Standard MIP Registration Protokoll durchgeführt werden müssen und dementsprechend die Kosten für die Entwicklung dieser Lösung vergleichsweise gering sind. Auf der anderen Seite wird das Protokoll dadurch weniger effizient, als es sein könnte, wenn man adäquate Änderungen zuließe.

Bei diesem Ansatz authentifizieren sich der Nutzer und der kontaktierte ISP gegenseitig. Danach wird die Authentifizierungsinformation zum RSP / HA weitergeleitet und von dort zum Unternehmen weitergeschickt. Danach erfolgt die MIP Registrierung. Der Ansatz ist in Abbildung 107 dargestellt.

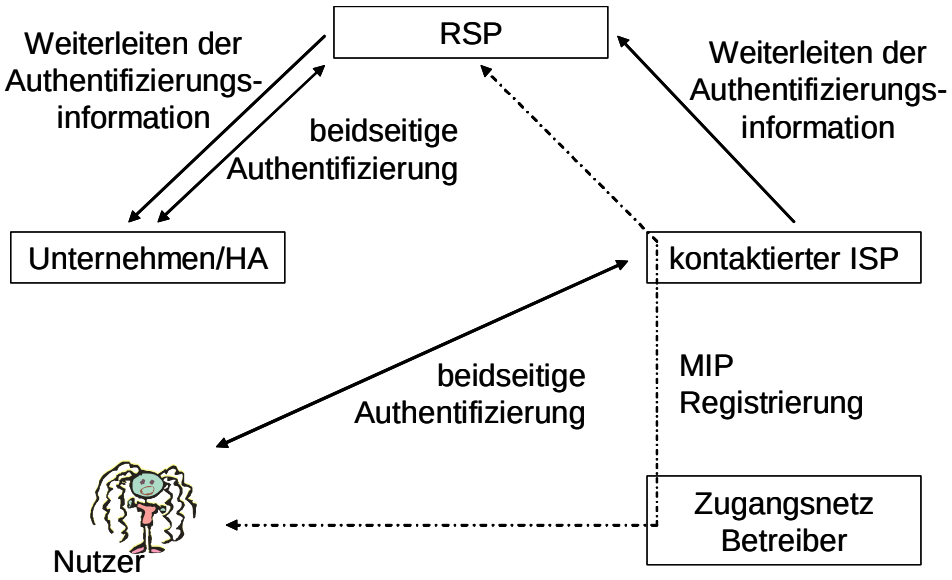


Abbildung 107: Separate Nutzer Authentifikation durch kontaktierten ISP und MIP Registration

Die erweiterte Notation aus Abschnitt 5.3.1 wird auch hier verwendet. Das Protokoll entspricht dem aus Abschnitt 5.3.1 abgesehen von der Tatsache, dass der Registrierungsprozess zwischen Nutzer U und dem RSP / HA abgewickelt wird anstatt zwischen U und dem HA im Unternehmen. Das Authentifikationsprotokoll ist in Abbildung 108 dargestellt.

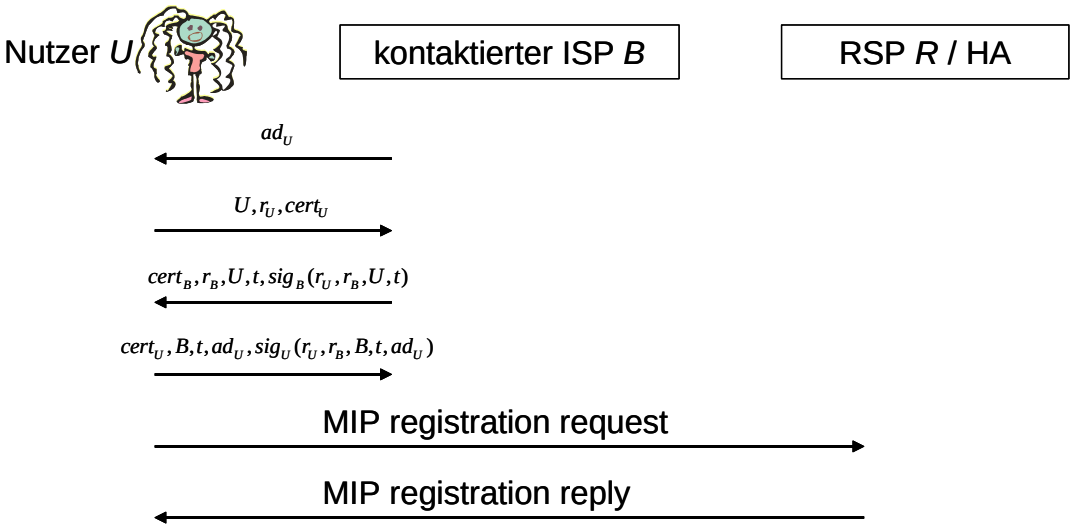


Abbildung 108: Protokolle für getrennte Nutzer Authentifikation durch kontaktierten ISP und MIP Registration

Die Weiterleitung der Authentifikationsinformation zeigt Abbildung 109.

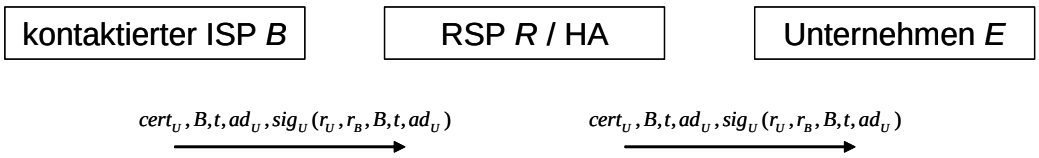


Abbildung 109: Weiterleiten der Authentifikationsinformation zum RSP und Unternehmen

Analog zu Abschnitt 5.3.1 bietet sich auch hier die Möglichkeit an, die Weiterleitung der Nutzerauthentifikationsinformation zu elimieren. Die Weiterleitung vom kontaktierten ISP zum HA beim RSP kann wegfallen, wenn dafür die entsprechende Information im Rahmen des MIP Registration Prozesses mitgeführt wird, wie in Abbildung 110.

Die eliminierte Nachricht wird durch eine Nachricht ersetzt, die innerhalb des MIP Registration Protokolls gesendet wurde. Dies erfordert die Einführung einer neuen Erweiterung im MIP Registration Protokoll, die der Standard nicht enthält. Allerdings würde eine derartige Modifikation eine effizientere Lösung bedeuten.

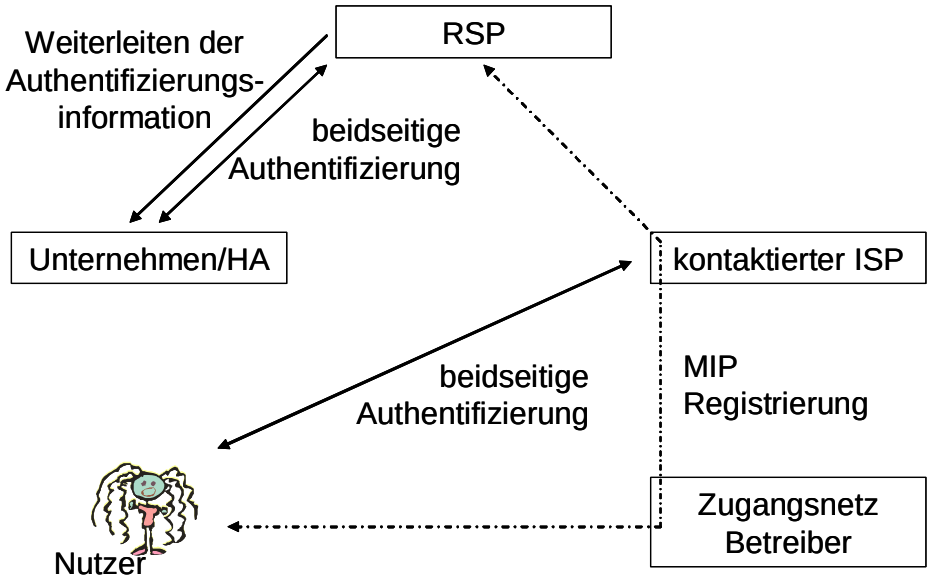


Abbildung 110: Elimination einer Weiterleitung der Nutzer Authentifikationsnachricht durch Ausbeutung des MIP Registrations-Protokolls

5.4.1.1 Delegation der Authentifikation des Nutzers

In diesem Abschnitt wird die Lösung für das Szenario der Delegation der Nutzerauthentifikation vom kontaktierten ISP zum RSP bei separater MIP Registration gezeigt. Sie ist in Abbildung 111 dargestellt. Das zugehörige Authentifikationsprotokoll zeigt Abbildung 112.

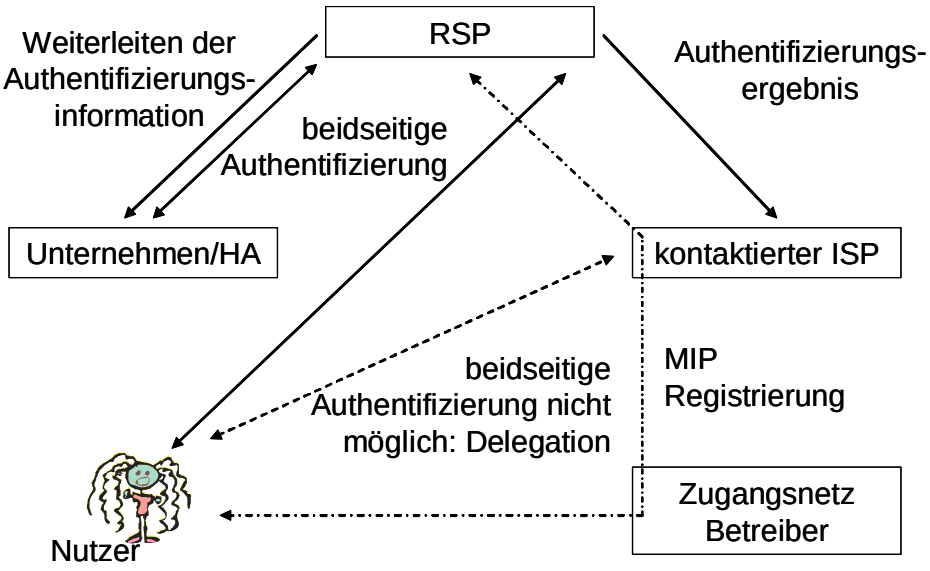


Abbildung 111: Delegation der Nutzer Authentifikation vom kontaktierten ISP zum RSP bei separater MIP Registration

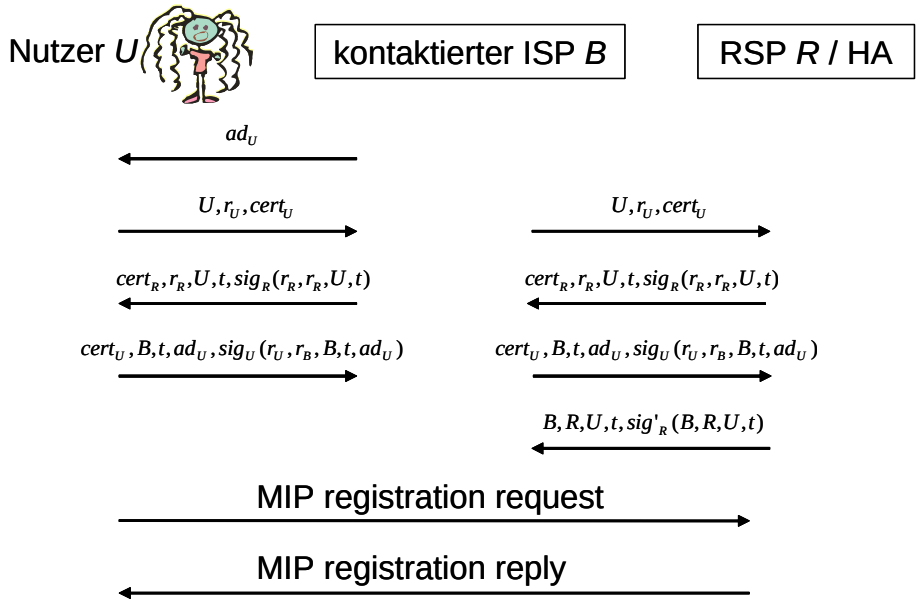


Abbildung 112: Protokolle für getrennte Authentifikationsdelegation und MIP Registration

5.4.2 Kombinierte Nutzerauthentifikation durch den kontaktierten ISP und den HA auf Basis einer modifizierten MIP Registration

In diesem Abschnitt wird eine Lösung gezeigt, welche die Nutzerauthentifikation mit der MIP Registration in einem Protokoll verbindet. Es wird wieder dieselbe Notation wie schon in Abschnitt 5.3.2 verwendet. Sie ist in Abbildung 113 dargestellt. Abbildung 114

zeigt das zugehörige kombinierte Protokoll für Authentifikation und MIP Registration. Abbildung 115 zeigt die Weiterleitung der Authentifikationsinformation.

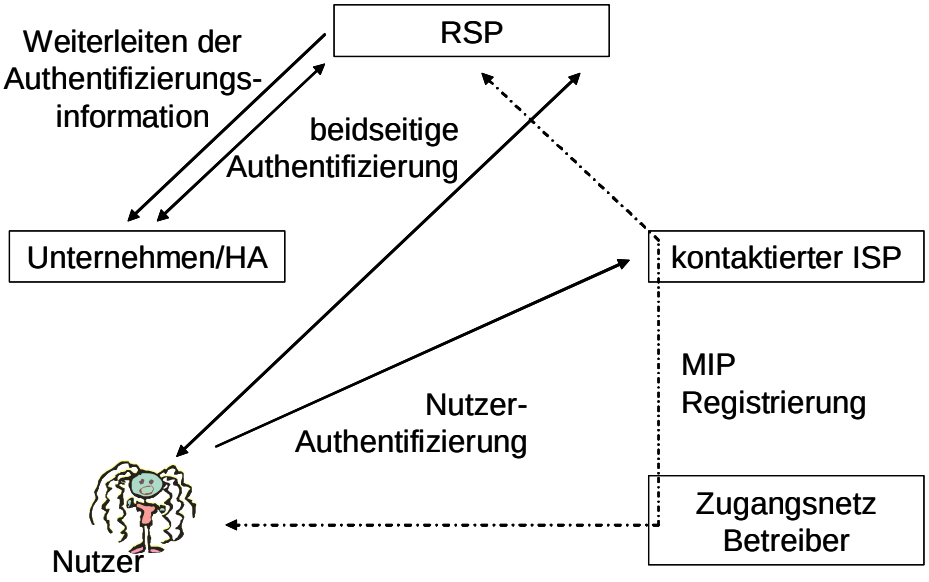


Abbildung 113: Kombinierte Nutzer Authentifikation und MIP Registration in einem Verifikations-Protokoll

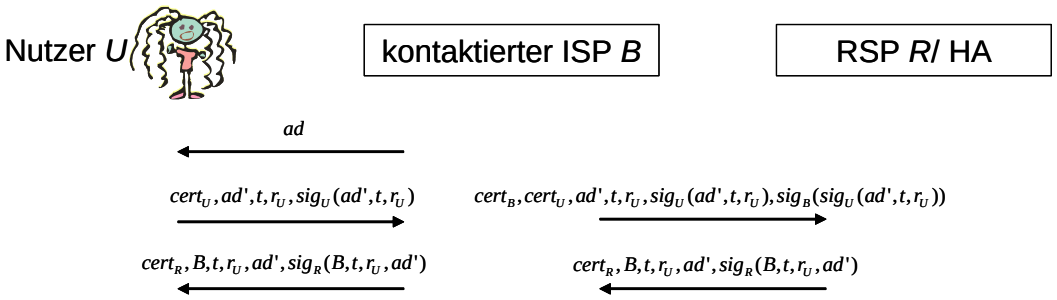


Abbildung 114: kombiniertes Protokoll für Authentifikation und MIP Registration

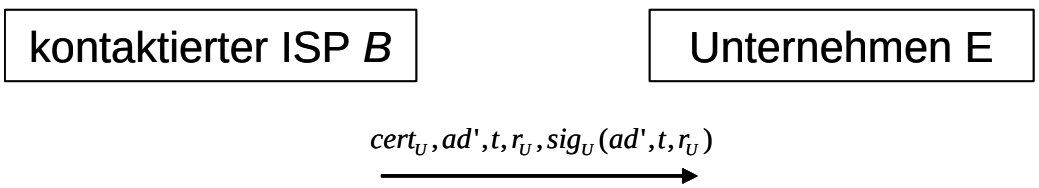


Abbildung 115: Weiterleitung der Nutzer Authentifikationsinformation

5.4.2.1 Delegation der Authentifikation des Nutzers

Die Lösung für die Delegation der Nutzerauthentifizierung, passend zum vorigen Abschnitt wird hier in Abbildung 116 dargestellt. Die Argumente und

Schlussfolgerungen, die in den vorhergehenden Abschnitten dargestellt wurden, finden auch hier Anwendung. Das entsprechende kombinierte Protokoll zur Authentifikation und MIP Registration ist in Abbildung 117 gezeigt.

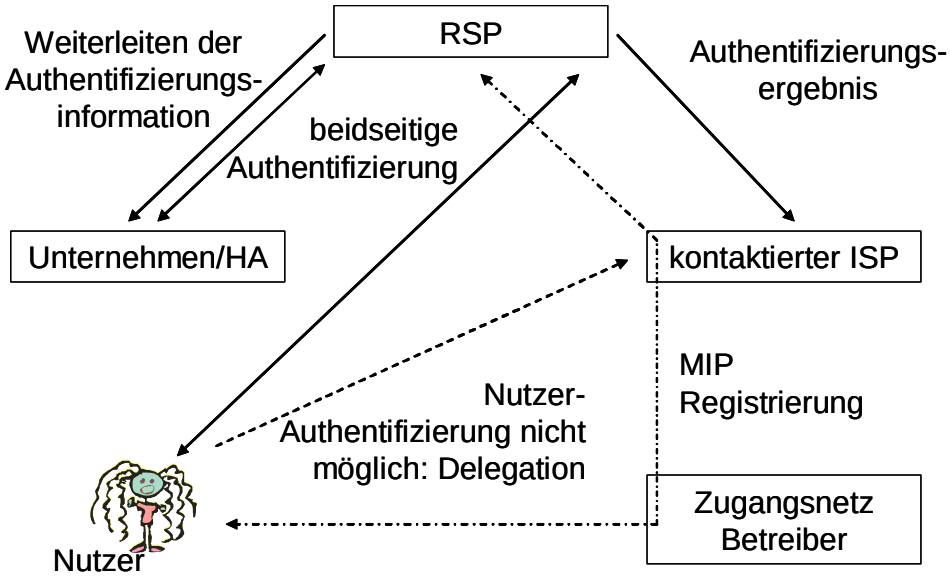


Abbildung 116: Kombinierte Nutzer Authentifikation und MIP Registration in einem Protokoll mit Delegation

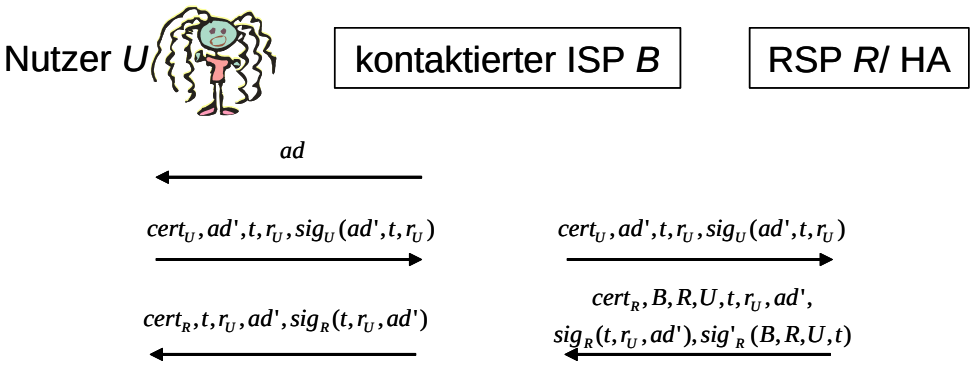


Abbildung 117: Das Protokoll für kombinierte Authentifikation und MIP Registration

Nachdem in diesem Kapitel eine generische Authentifizierungslösung aufgezeigt wurde, wird im nächsten Kapitel eine PKI-basierte Lösung entwickelt.

6 PKI basierte Lösung

Das in Kapitel 3 vorgestellte RSP Geschäftsmodell zielt auf Authentifizierungsmechanismen ab, die auf Public Key Kryptographie und den Einsatz von Identitätszertifikaten für Personen und Maschinen erfordern. Wenn man Public Key Kryptographie einsetzen will, dann braucht man eine entsprechende Infrastruktur. Im Allgemeinen beschäftigt sich eine Public Key Infrastruktur (PKI) mit vielen Aspekten bezüglich des Einsatzes von öffentlichen Schlüsseln. In dieser Arbeit wird hauptsächlich der Aspekt der Erzeugung und Ausstellung von Public Key Zertifikaten und der Bereitstellung von Informationen bezüglich der Gültigkeit dieser Zertifikate betrachtet.

Wenn für eine Entität ein Zertifikat ausgestellt wird, dann bescheinigt der Aussteller des Zertifikates, dass ein bestimmter öffentlicher Schlüssel zu einer ganz bestimmten Entität gehört. Der Aussteller selbst verfügt ebenfalls über einen öffentlichen Schlüssel, dessen Zugehörigkeit in einem Zertifikat bestätigt wird. Dieses Zertifikat kann von einem anderen Aussteller ausgestellt sein oder von ihm selbst. In letzterem Fall spricht man von einem selbst signierten Zertifikat.

Jedes Zertifikat verfügt über eine vordefinierte maximale Gültigkeitsdauer. Es ist jedoch möglich, dass ein Zertifikat vor Ablauf seiner vordefinierten Gültigkeit zurückgerufen wird. Daher muss jedes Mal, wenn auf Basis eines Zertifikates eine Entscheidung getroffen werden soll, der aktuelle Status des Zertifikates überprüft werden.

In den RSP Geschäftsmodellen wird eine zertifikatebasierte Authentifizierung zum Zweck der späteren Autorisierung und Abrechnung eingesetzt. In diesem Kontext ist es notwendig, dass die Sperrung von Zertifikaten sehr schnell und effektiv geschieht. Die Sperre von Zertifikaten kann verschiedene Gründe haben. Ein Grund kann sein, dass ein Nutzer seinen geheimen Schlüssel „verloren“ hat oder dieser irgendwie in falsche Hände gelangt ist. Wenn der geheime Schlüssel eines Nutzers ausspioniert wurde, hat der Nutzer ein Interesse zu verhindern, dass sein öffentlicher Schlüssel und das zugehörige Zertifikat von einem Widersacher verwendet werden kann, um Dienste unter seiner Identität zu nutzen. Des Weiteren hat ein ISP ein Interesse daran, dass ein Nutzer, der seinen Vertrag kündigt, nicht weiter unentgeltlich seine Dienste Nutzen kann. Dafür muss er in der Lage sein, das Nutzerkonto sofort zu blockieren. Das Blockieren von Benutzerkonten sollte auch schnell und effektiv möglich sein, wenn ein mobiler Nutzer zwischen verschiedenen ISPs hin- und herwandert. Er könnte versuchen, bei einem ISP, mit dem er selbst keinen Vertrag hatte, auf Basis seines alten gekündigten Vertrages Dienste zu nutzen. Die Situation ist ähnlich, wenn ein Angestellter eines Unternehmens auf Grund der Veränderung seiner Position im Unternehmen ein anderes Zertifikat benötigt. In dem Fall, dass der Angestellte das Unternehmen verlässt, hat das Unternehmen i. d. R. ein Interesse daran, dass der ehemalige Angestellte die mit seiner Unternehmenszugehörigkeit verbundenen mobilen Kommunikationsdienste nicht mehr nutzen kann. Dies bedeutet,

dass die Überprüfung des Status des Zertifikates immer wichtig ist, wenn die Identität eines Nutzers überprüft werden soll.

Im Allgemeinen beinhaltet die Überprüfung des Status eines Zertifikates die Überprüfung einer ganzen aus vielen Zertifikaten bestehenden Zertifikatskette. Die Überprüfung des Status von allen Zertifikaten in der Kette macht die Konstruktion eines entsprechenden Zertifikatspfades notwendig. Danach müssen die Signaturen in allen Zertifikaten überprüft werden und schließlich müssen alle Zertifikate hinsichtlich ihres Sperr-Status überprüft werden. In Abhängigkeit von der zugrunde liegenden Zertifikatshierarchie, kann dies ein sehr komplexer und daher auch sehr zeitraubender Vorgang sein.

Zusätzliche Schwierigkeiten bereitet die Bereitstellung von aktuellen, vertrauenswürdigen und überprüfbaren Informationen hinsichtlich des Status der Zertifikate von denjenigen Parteien, die vom mobilen Nutzer authentifiziert werden müssen wie z. B. Zugangsnetzbetreiber oder ISP. Dies ist ein schwieriges Problem, da der mobile Nutzer während des Authentifizierungsvorgangs zunächst keinen Netzzugang hat, um Informationen z. B. über den Status von Zertifikaten zu erhalten.

Im Rahmen dieser Arbeit wird ein spezifisches Modell einer PKI Infrastruktur für das Roaming zwischen heterogenen Netzen im Sinne der in Kapitel 3 beschriebenen Geschäftsmodelle entwickelt. Die hier vorgeschlagene PKI Infrastruktur erfüllt die Anforderung, dass der Aufwand, der benötigt wird, den Status von Zertifikaten zu überprüfen, gering ist. Z. B. wird die maximale Länge der möglichen Zertifikatsketten auf einen kleinen Wert begrenzt. Zusätzlich wird für den Fall von verschiedenen RSP Domänen u. a. für einen mobilen Nutzer, der zwischen unterschiedlichen Netzen umherwandert, die zu unterschiedlichen RSP Domänen gehören, betrachtet. Der Nutzer kann so auch zu ISPs, die mit einem anderen RSP kooperieren, als mit dem, mit dem der ISP kooperiert, mit dem der umherwandernde Nutzer einen Vertrag abgeschlossen hat. Die hier vorgeschlagene PKI Struktur integriert darüber hinaus eine bestehende Unternehmens-PKI. Des Weiteren wird eine neue X.509v3 Zertifikatserweiterung vorgeschlagen, welche die Unternehmens CAs verwenden sollten, um die Konstruktion der Zertifikatspfade für die Zertifikate ihrer mobilen Mitarbeiter zu unterstützen. Weiter ist es möglich, die Statusüberprüfung zu PKI Servern zu delegieren. PKI Server sind Server bzw. Komponenten, welche auf die Überprüfung von Zertifikaten und damit zusammenhängende Vorgänge spezialisiert sind. Diese können von RSPs oder ihnen assoziierten Unternehmen bereitgestellt werden. Die Funktionsweise der PKI Server im Zusammenspiel mit den anderen Komponenten der Roaming Architektur wird hier erklärt. Der Einsatz der PKI Server erlaubt es dritten Parteien, die komplette Arbeit der Verifikation von Zertifikaten zu delegieren. Die Möglichkeit der Delegation ist extrem attraktiv beim Einsatz mobiler Endgeräte, welche stark begrenzte Kapazitäten haben und welche die im Rahmen des Verifikationsprozesses anfallende Arbeit nicht in vertretbarer Zeit ausführen können. Zusätzlich wird die im Rahmen dieser Arbeit entwickelte Lösung für das oben angeschnittene Problem eines mobilen Nutzers beschrieben, der ohne Netzzugang das Zertifikat der von Ihm kontaktierten Partei überprüfen will.

Nachdem in Kapitel 2.5 die Grundlagen einer PKI erklärt worden sind, werden in Abschnitt 6.1 die bereits beschriebenen Komponenten hinsichtlich ihrer Aufgaben in der hier entwickelten Architektur und die hier benötigten Mechanismen beschrieben. Kapitel 6.2 stellt PKI relevante Standards vor, die möglicherweise für die hier vorgeschlagene Lösung eingesetzt werden können. In Abschnitt 6.3 wird dann die Lösung für die Roaming Architektur dargestellt. In Abschnitt 6.4 wird die Lösung für die Überprüfung der Zertifikate bei der Authentifizierung hinsichtlich ihres Status beschrieben, wobei verschiedene Fälle, wie für Mitarbeiter eines Unternehmens oder private Nutzer, die keinem Unternehmen angehören sowie Zugehörigkeit zu unterschiedlichen Domänen, betrachtet werden. In Abschnitt 6.5 wird dann die umgekehrte Richtung also die Überprüfung der Zertifikate der Parteien, die vom mobilen Nutzer überprüft werden betrachtet. Hierbei wird das Problem des Nutzers, der ein Zertifikat überprüfen will, ohne einen Internetzugang zu haben, ausführlich behandelt.

6.1 Grundlegende PKI Funktionalität

Zunächst werden im Folgenden die in 2.5 allgemein aufgeführten Komponenten hinsichtlich ihrer Aufgaben in der hier entwickelten Architektur und die hier benötigten Mechanismen beschrieben.

6.1.1 Certification Authority

Eine CA im weitesten Sinne ist eine Entität, die beliebige Arten von Zertifikaten verteilt. Diese Arbeit beschränkt sich auf solche Zertifikate, in denen die Zugehörigkeit eines öffentlichen Schlüssels zu einer Person bestätigt wird. Mit ihrer Signatur im Zertifikat bestätigt die CA einer Entität den Besitz bzw. die Zugehörigkeit eines öffentlichen Schlüssels. Es kann hier zwischen öffentlichen CAs, Unternehmens CAs, CAs, die einem bestimmten Dienst assoziiert sind, wie z. B. eine CA, die Zertifikate für Kunden eines bestimmten ISPs ausstellt, oder CAs mit spezifischer Funktionalität innerhalb einer hierarchischen PKI, wie z.B. einer „root“ oder „top level“ CA, unterschieden werden.

Neben der Erzeugung von Zertifikaten sind CAs auch verantwortlich für die Erzeugung von Sperrinformationen. Dafür kann eine CA CRLs, Delta-CRLs[24], oder einen OCSP Dienst für die Bereitstellung von Statusinformation für einzelne Zertifikate bereitstellen.

CAs können in einer PKI Hierarchie liegen, was eine Zertifikatshierarchie, impliziert. Eine Zertifikatshierarchie meint eine baumartige Topologie von Zertifikatsbeziehungen zwischen CAs. Möglicherweise existieren verschiedene PKI Hierarchien nebeneinander her. Da CAs Zertifikate sowohl für End-Nutzer als auch für andere CAs ausstellen können, kann es nötig sein, die Zertifikatspfade zu validieren, um den Status eines Nutzerzertifikates zu erhalten. Es ist möglich, dass Hierarchien durch CAs verbunden sind, die über Kreuz von anderen Hierarchien zertifiziert sind. So können die Zertifikatspfade, welche zur Validierung aufgebaut werden, unterschiedliche Hierarchien überspannen.

[24] Siehe Abschnitt 6.1.4

6.1.2 Registrierungsautorität

Eine Registrierungsautorität (RA) führt die Registrierung der End-Entitäten durch. Dafür muss die RA die notwendigen relevanten Daten von den Entitäten sammeln. Darüber hinaus, ist die RA dafür verantwortlich zu überprüfen, ob die gesammelten Daten korrekt sind. Die Intensität und der Aufwand dieser Verifikation können unterschiedlich sein und sollten von den Möglichkeiten bzw. der Mächtigkeit des zu erzeugenden Zertifikates abhängen. Während des Vorgangs der Registrierung führt die RA Überprüfungen der Identität durch. Die Prozeduren entsprechen der festgelegten Strategie der CA. Die RA kann entweder von der Partei betrieben werden, die selbst die CA betreibt oder auch von einer beliebigen anderen Partei. In dem Fall, dass die RA und die CA von unterschiedlichen Parteien betrieben werden, liegt ihrer Kooperation ein bestimmtes Vertrauensverhältnis zu Grunde. Dies kann variieren vom dem Fall, in welchem die CA der RA dahingehend vertraut, dass sie die Daten verlässlich auf Korrektheit überprüft hat, bis zu dem Fall, in dem die CA sich nicht darum kümmert, dass die Daten korrekt sind, bzw. sich nicht dafür interessiert, ob die Daten korrekt sind.

6.1.3 Über-Kreuz-Zertifizierung

Ohne Über-Kreuz-Zertifizierung ist die Überprüfung der Zertifikate nur innerhalb derselben PKI Domäne möglich. Über-Kreuz-Zertifikate erlauben Entitäten, wie z. B. Nutzern oder beliebigen Maschinen Zertifikate anderer Entitäten, die von anderen Parteien aus anderen Hierarchien ausgestellt wurden, zu validieren. Für gewöhnlich stellt eine CA ein Kreuz-Zertifikat für eine andere CA aus, die in eine andere Hierarchie eingebunden ist. Kreuz-Zertifizierung kann entweder unidirektional oder bidirektional sein. Abbildung 118 stellt zwei CAs, welche zu unterschiedlichen Hierarchien gehören, dar, die sich gegenseitig zertifiziert haben, was einer bidirektionalen Über-Kreuz-Zertifizierung entspricht.

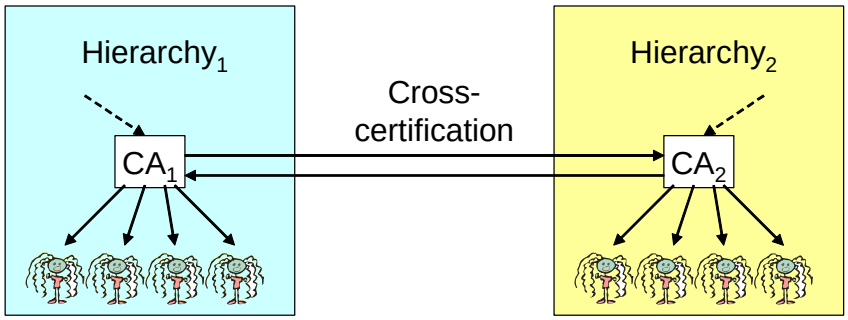


Abbildung 118: Gegenseitige Über-Kreuz-Zertifizierung

6.1.4 Sperre von Zertifikaten

Wenn ein Zertifikat vor seinem Verfallsdatum nicht mehr benutzt werden soll, dann muss es zurückgerufen („revoked“) bzw. gesperrt werden. Dafür kann es die verschiedensten Gründe geben. Dies sind z. B.:

- Kompromittierung des Schlüssels,

- Kompromittierung der CA,
- Änderung der persönlichen Daten des Zertifikatsinhabers
- Änderung der Beschäftigung im Falle eines für einen Unternehmensmitarbeiter ausgestellten Zertifikates,
- Änderung des Kundenstatus im Falle eines auf einen Dienst bezogenen Zertifikates oder
- Änderung einer Erlaubnis, wenn diese im Zertifikat enthalten ist.

Wenn eine Entität durch Anwendung eines Protokolls, in dem Zertifikate eingesetzt werden, identifiziert wird, genügt es nicht, die Signatur des Zertifikats der Entität und vielleicht noch die Signatur der Zertifikate einiger CAs zu überprüfen. Darüber hinaus ist es erforderlich, dass der Status des Zertifikates bezüglich einer möglichen Sperre überprüft wird. Dies setzt voraus, dass die CAs periodisch aktuelle Informationen über den Sperrstatus der von ihnen erzeugten Zertifikate veröffentlichen. In der Praxis sollten neue Sperrinformationen innerhalb von Intervallen, die wenige Stunden umfassen - z. B. 8 Stunden -, bereitgestellt werden. Je kleiner diese Intervalle sind, desto kleiner ist der potentielle Schaden bzw. der Verlust, der durch den Missbrauch solcher Zertifikate verursacht werden kann. Für gewöhnlich sollten Zertifikate, wenn sie ausgestellt werden, eine Referenz zu einer Adresse beinhalten, an der die betreffende Sperrinformation zu finden ist.

Es gibt verschiedene Möglichkeiten für eine CA, die notwendige Information für andere Parteien bereitzustellen:

- **CRLs**: Bei CRLs handelt es sich um signierte Listen, welche alle zurückgerufenen Zertifikate enthalten, deren im Zertifikat angegebene Gültigkeitsdauer nicht überschritten ist. Wenn das Verfallsdatum eines Zertifikates erreicht ist, wird das Zertifikat nicht mehr in der nächsten aktualisierten CRL aufgeführt. CRLs können bereitgestellt werden als Files, welche eine signierte Liste aller zurückgerufenen Zertifikate beinhaltet. Dies bedeutet, dass die Download-Zeit in Abhängigkeit von der Größe der Datei variiert. Des Weiteren muss die gesamte Datei heruntergeladen werden, auch wenn der Verifizierer nur die Status Information eines einzigen Zertifikates benötigt. Ein weiterer Nachteil der CRLs ist, dass es Zeitintervalle gibt, in denen die Information über den Status der Zertifikate nicht aktuell ist. Je nachdem, in welchen Intervallen die CRLs aktualisiert werden, ist die Gefahr gesperrte Zertifikate in Unkenntnis der Sperre nicht zurückzuweisen mehr oder weniger groß. Das bedeutet, dass die Aktualität der Sperrinformation, die potentiellen Verifizierern zugänglich gemacht wird, völlig von der Häufigkeit abhängt, mit der die Sperrlisten (CRLs) aktualisiert werden. Je kleiner die Zeitintervalle sind, desto besser sind die Verifizierer informiert und desto teurer ist der Vorgang.
- **Delta-CRLs**: Delta-CRLs stellen Daten bereit über die Veränderungen bezüglich der Sperrinformationen. Sie enthalten ausschließlich Sperrinformationen über die Zertifikate, die nach dem Bereitstellen bzw. Verteilen der letzten Zertifikats-Sperrlisten zurückgerufen wurden. Man erhält die aktuelle CRL also immer, indem man die neue Delta-CRLs der alten CRL hinzufügt. Wenn man den

Sperrstatus eines Zertifikates überprüfen will, muss man immer alle Delta-CRLs zusammen mit der ersten CRL überprüfen. Im Vergleich zu herkömmlichen CRLs bedeuten Delta-CRLs eine Kostenreduzierung dadurch, dass weniger Datenvolumen heruntergeladen werden muss und dementsprechend eine kürzere Zeit für die Datenübertragung benötigt wird. Beim Transfer herkömmlicher CRLs wird sehr viel Information redundant übertragen. Bei kurzen Zeitintervallen, in denen wenige Veränderungen vorliegen, sind die Redundanzen besonders groß. Im Extremfall wird wenn keine Veränderungen vorliegen zweimal die gleiche CRL übertragen. Delta-CRLs vermeiden diese Redundanzen, da nur neue Informationen über die Zertifikate, deren Status sich im letzten Zeitintervall verändert hat, übertragen werden. Wie auch bei herkömmlichen CRLs ist der Nachteil der Delta-CRLs, dass die Aktualität und damit der Wert der Information stark von dem Zeitintervall abhängt, in dem sie veröffentlicht werden.

- **OCSP**: Das Online Certificate Staus Protocol (OCSP) erlaubt es, den gegenwärtigen Sperrstatus einzelner spezifischer Zertifikate abzufragen. Dies kann die Kosten sowohl, was das Datenvolumen als auch die für die Datenübertragung benötigte Zeit angeht, reduzieren. Des Weiteren löst dieser Ansatz das Problem der Aktualität der Daten über die Statusinformation, da immer zu dem Zeitpunkt, an dem sie benötigt werden, gezielt die aktuellen Daten abgefragt werden. Das Problem der Zeitintervalle entfällt somit.

Es sei hier noch angemerkt, dass die oben beschriebenen Mechanismen für selbst-signierte Zertifikate nicht angewandt werden können. Für derartige Zertifikate werden andere Mechanismen als CRLs oder OCSP benötigt.

6.1.5 Pfad Validierung

Ein Zertifikatspfad enthält den Weg von einem gegebenen ersten Zertifikat – dem eigentlich zu überprüfenden Zertifikat - zu einer CA aus einer vordefinierten Menge von CAs. Diese Menge wird normalerweise von der überprüfenden Partei vordefiniert. Die Elemente dieser Menge bezeichnet man als Vertrauensanker. Die Auswahl solcher Vertrauensanker ist eine Frage der zugrunde liegenden Strategie der überprüfenden Partei. In einer hierarchischen PKI könnte dies die CA auf der obersten Ebene sein. Die CA, welche das Zertifikat der überprüfenden Partei ausgestellt hat, ist typischerweise auch ein Vertrauensanker für diese Partei. Prinzipiell kann jede existierende CA auch als Vertrauensanker betrachtet werden. Dies hängt allein von der Strategie der jeweiligen Partei ab, welche CAs sie als vertrauenswürdig betrachtet und zu Vertrauensankern macht.

Im Allgemeinen kann ein Zertifikatepfad bezogen auf einen bestimmten Zeitpunkt, d.h. Datum und Uhrzeit, validiert werden. Gewisse Rückschlüsse, wie lange in der Vergangenheit dieser Pfad schon gültig ist, lassen sich auch ziehen. Ob dieser Pfad zu einem in der Zukunft liegenden Zeitpunkt noch gültig ist bzw. wie lange er gültig sein wird, lässt sich aber nicht mit Bestimmtheit voraussagen. Wenn a posteriori eine Verifikation von Authentifizierungsdaten notwendig ist, z. B. für den Zweck nicht abstreitbarer Abrechnung, kann die Validierung eines Zertifikatspfades für einen in der

Vergangenheit liegenden Zeitpunkt erforderlich sein. Die Validierung eines Zertifikatspfades beinhaltet iterativ die Überprüfung jedes einzelnen Zertifikates innerhalb des Pfades bis zum Vertrauensanker hinauf. Formal ausgedrückt bedeutet dies: Bei einem gegebenen Zertifikat c_n beinhaltet der Prozess der Validierung des Pfades, die Verifizierung dessen, dass bei einer Reihenfolge von Zertifikaten $c_1,\ldots,c_n$ das Subjekt des Zertifikates c_i der Aussteller des Zertifikates c_{i+1} ist für $i=1,\ldots,n-1$. Es bedeutet weiterhin, dass alle Zertifikate $c_1,\ldots,c_n$ für den Zeitpunkt, für den die Überprüfung erforderlich ist, gültig sein müssen. Der Vertrauensanker ist der Aussteller des Zertifkates c_1.

Voraussetzung für die Validierung eines Zertifikates ist, dass ein Pfad vom zu überprüfenden Zertifikat zu einem Vertrauensanker gefunden wird. Jedes auf diesem Pfad liegende Zertifikat muss dann einzeln für sich überprüft werden. Diese Prozedur beinhaltet:

- Das kryptographische Überprüfen der Signatur,
- das Überprüfen von Inhalten des Zertifikates, wie z.B. die Gültigkeitsdauer und
- die Überprüfung des Status des Zertifikates.

6.2 PKI Standards

6.2.1 Internet X.509 Public Key Infrastructure

Die Struktur von X.509 Zertifikaten und Sperrlisten sowie deren Anwendung ist im RFC 3280 beschrieben [HouPolForSol02]. RFC 3280 liefert das Format, die Semantik der Zertifikatselemente und das Sperrprofil für die Internet PKI beschrieben. Weiter: werden Lösungen für das Bearbeiten von Zertifikatspfaden im Internet gegeben. Zertifikatestrukturen und deren Erweiterungen sind in der Version 3 (X.509v3) festgelegt. Die Abbildung 119 zeigt die Struktur eines X.509v3 Zertifikats. Diese Struktur besteht aus Basisfeldern sowie Standard- und Privat-Erweiterungen. Die Erweiterungen unterstützen die Erzeugung einer kompatiblen und wiedernutzbaren Internet PKI. Die Standarderweiterungen werden für die Einbindung von zusätzlichen Attributen zum Nutzer und für die Einrichtung von Zertifikatehierarchien benutzt, welche für die Pfadvalidierung benötigt werden. Die Standarderweiterungen entsprechen den X.509v3 Zertifikaten. Privaterweiterungen können für bestimmte nur spezifische Gruppen betreffende Informationen benutzt werden. RFC 3280 definiert Privaterweiterungen, welche die PKI Internet Gemeinde unterstützen. Im Allgemeinen sind Erweiterungen mit „kritisch“ oder „nicht-kritisch“ gekennzeichnet, was für deren Verarbeitung wichtig ist. Wenn ein System eine Erweiterung nicht erkennt, welche kritisch ist, muss es das Zertifikat zurückweisen. Das Zertifikat muss ebenfalls zurückgewiesen werden, wenn die Erweiterung erkannt wird, aber deren Verifikation nicht möglich ist. Nicht-kritische Erweiterungen dürfen dagegen auch wenn die Verifikation fehlschlägt ohne Zurückweisung ignoriert werden.

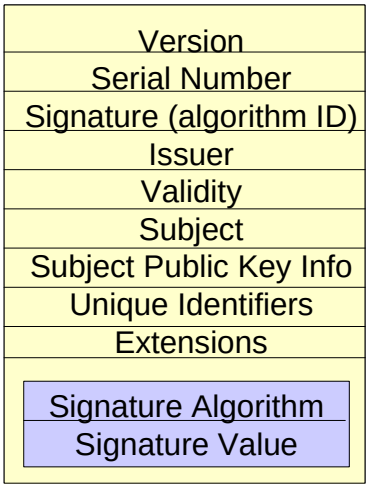


Abbildung 119: Struktur eines X.509v3 Zertifikates

RFC 3280 beschreibt die CRL.Version 2, welche eine allgemeine Grundlage für die Zusammenarbeit bei Anwendungen ist. Weiter werden Richtlinien für die Nutzung der Erweiterungen gegeben und Annahmen und Forderungen hinsichtlich der CRL Informationen beschrieben.

Zertifikatepfade vom gleichen Vertrauensanker zu der gleichen Endinstanz könnten nicht für alle Anwendungen geeignet sein. Diese Begrenzungen für die Konstruktion von Zertifikatepfaden sind durch geeignete Zertifikateerweiterungen berücksichtigt, d.h. sie werden in *„basic constraints“* und *„policy constraints“* Erweiterungen ausgedrückt. Der Pfadvalidierungsprozess beachtet diese Limits aller Zertifikate im Pfad. Zusätzliche Angaben hinsichtlich der Pfadlogik sind für das Verarbeiten des Zertifikatepfads von Vorteil.

6.2.2 Das einfache Zertifikat-Validierungsprotokoll (SCVP)

Der Prozess einer Zertifikatvalidierung kann sehr komplex sein. In Abhängigkeit von der Länge des Zertifikatepfades kann er einen hohen Rechenaufwand erfordern. Bevor Zertifikatepfade validiert werden können, sind sie zu konstruieren, d.h. dass für jedes Zertifikat im Validierungspfad der aktuelle Status vom Verifizierer anzufordern ist. Dies muss von allen Teilnehmern erfolgen, die Zertifikate verifizieren, auch wenn einige von ihnen dieselben Zertifikate in nahezu der gleichen Periode schon einmal verifiziert haben. Der Rechenaufwand kann dabei reduziert werden, wenn es möglich ist, schon konstruierte Zertifikatepfade und ggf. deren Validationsergebnisse erneut zu verwenden. Um den Aufwand für die Konstruktion der Zertifikatepfade und deren Validierung zu reduzieren, werden spezielle PKI Server eingesetzt, welche die anderen Teilnehmer unterstützen. PKI Server arbeiten bei der Erzeugung der Zertifikatekette und bei deren Validierung viel effizienter. Ein solcher PKI Server kann die Konstruktion von Zertifikateketten und die Validierungsergebnisse erneut verwenden, was den gesamten Validierungsprozess für jene Teilnehmer beschleunigt, die eine Zertifikateverifikation erfordern. Weiter gestattet dieser Ansatz kleine Geräte mit beschränkter Rechenleistung. Solche Endgeräte können

selbst keine längeren Zertifizierungen in einer annehmbaren Zeit durchführen. Der Einsatz von PKI Servern wurde bereits in verschiedenen Publikationen, wie [HuFi02, Hunt02, MAMG99] vorgeschlagen.

Wenn ein Verifizierer die Dienste eines PKI Servers zur Unterstützung des Validierungsprozesses einsetzt, ist ein standardisiertes Protokoll erforderlich, welches die Interaktionen zwischen dem Verifizierer als Client und dem PKI Server festlegt. Ein solches Protokoll ist das *Simple Certificate Validation Protocol* (SCVP). Im Folgenden wird eine kurze Zusammenfassung des SCVP gegeben. Eine ausführlichere Beschreibung findet man [MaHF03]. Das SCVP definiert ein Protokoll zwischen einem Client und einem SCVP Server für die Delegation der Konstruktion der Zertifikatekette und deren Validierung. SCVP hat den Vorteil, diese Aufgabe für den Nutzer einfach zu machen. Weiter erlaubt SCVP die zentrale Verwaltung der PKI Strategien, welche auf spezielle Nutzer angewandt werden. Dies ist relevant, wenn Validierungsstrategie für die Zertifikate nicht vom prüfenden Client vorgegeben ist, sondern von einem anderen Teilnehmer, z.B. wenn Unternehmen die Strategien der Validierung der Zertifikatekette für Mitarbeiter festlegt.

SCVP Server bieten Dienste wie die Erzeugung von Zertifikateketten, Zertifikatevalidierungen und Zustandsinformationen über Zertifikate an. Die Dienste, die der Nutzer vom SCVP Server fordert, hängen vom Vertrauen ab, das der Nutzer dem SCVP Server entgegenbringt. Wenn der Nutzer dem SCVP Server nicht traut, versorgt dieser ihn nur mit Informationen, z.B. über die Pfadkonstruktion, aber diese selbst wird vom Nutzer durchgeführt. Wenn der Nutzer dem Server traut, wird er alle Dienste in Anspruch nehmen einschließlich der Zertifikatevalidierung.

SCVP ist ein Anfrage/Antwort-Protokoll. Es gibt zwei Anfrage- und Antwort-Paare: Das Paar für die Validierung von Anfrage/Antwort und das für die Validierung der Anfrage/Antwort-Strategie. Das Anfrage/Antwort-Strategie-Validierungsprotokoll ist optional. Wenn es gewählt wird, sollte es vor dem Anfrage/Antwort-Validierungsprotokoll ausgeführt werden. Abbildung 120 zeigt das SCVP-Anfrage/Antwort-Validierungsprotokoll schematisch.

Die Validierungsanfrage wird vom Teilnehmer benutzt, vom SCVP Server einen gewünschten Dienst anzufordern. Die Validierungsanfrage kann signiert sein oder nicht. Die Signatur kann vom SCVP Server dazu benutzt werden, den anfragenden Teilnehmer zu authentifizieren. Die Validierungsanfrage enthält die gesamte Information über das dazugehörende Zertifikat hinsichtlich des angeforderten Dienstes vorausgesetzt, diese Information ist für den SCVP Server verfügbar. Das Ergebnis des angeforderten Dienstes ist entweder einen Zertifikatepfad zu einer Wurzel als Vertrauensinstanz für die in der Anfrage enthaltenen Zertifikate zu konstruieren oder einen validierten Zertifikatepfad zu einer Vertrauensinstanz aufzubauen oder einfach den Status für die Zertifikatekette zu liefern. Im Falle der Statusanfrage kann der Client den Server um Überprüfung des Zertifikatestatus bitten oder er kann um einen vollständigen Informationssatz nachsuchen, wie z.B. die CRLs, so dass er selbst den Zertifikatestatus überprüfen kann. Zusätzlich

kann die Art der Information, die vom SCVP Server an den SCVP Client als Antwort gegeben wird, spezifiziert werden. Außerdem kann der SCVP Client eine Strategie an den SCVP Server geben, welche dieser bei der Validierung beachten muss. Außer dem Vorschreiben der Validierungsstrategie, kann der SCVP Client direkt die Vertrauensanker vorgeben, die vom SCVP Server zu benutzen sind. Eine andere Möglichkeit erlaubt es dem SCVP Client, den Zeitpunkt festzulegen, für den der SCVP Server den Status überprüfen soll, wenn z. B. eine Partei am Zertifikatestatus zu einem Zeitpunkt in der Vergangenheit interessiert ist.

Die Validierungsantwort vom SCVP Server kann signiert sein oder nicht. Eine unsignierte Antwort muss nur zur Vermeidung eines Fehlerstatus generiert werden. Wenn die SCPV Antwort signiert ist, muss sie das Zertifikat des Servers enthalten. Jede Antwort enthält eine Referenz, die eindeutig die Anfrage identifiziert, zu der die Antwort gehört. Neben anderen Informationen liefert der SCVP Server die Ergebnisse des angeforderten Dienstes an den Client, z. B. den Antwortstatus, die Validierungszeit, die Validierungsstrategie, die Ergebnisse der Zertifikatevalidierungen wie „gut“, „zurückgewiesen“, „unbekannt“, „nicht vorhanden“.

Die Anfrage nach der Validierungsstrategie kann vom Client dazu benutzt werden, alle Validierungsstrategien des SCVP Servers zu erfragen. Die entsprechende Information wird dem Teilnehmer in der Antwort zur Anfrage nach der ValidierungsStrategie gegeben.

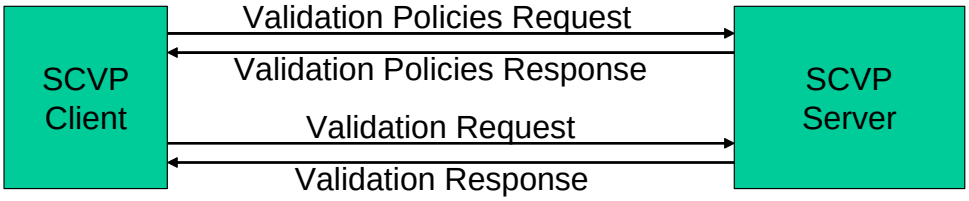


Abbildung 120: SCVP „Request“ und „Response“

6.2.3 Online Certificate Status Protocol (OCSP)

Das standardisierte *Online Certificate Status Protocol* (OCSP) erlaubt in Echtzeit die Überprüfung eines Zertifikatestatus ohne Sperrlisten anzufordern. OCSP bietet somit eine bessere Möglichkeit, verglichen mit der Sperrlisten-Variante, rechtzeitig Informationen über den Sperrzustand eines Zertifikates zu erhalten. Die Ursache liegt darin, dass zwischen der Sperrung eines Zertifikats und der Verbreitung der entsprechenden Sperrliste ein gewisser Zeitraum vergeht. Ein weiterer Vorteil des Online Certificate Status Protocol (OCSP) ist das verglichen mit dem Herunterladen kompletter Sperrlisten beträchtlich reduzierte übertragene Datenvolumen.

Das OCSP Protokoll ist ebenfalls ein Anfrage/Antwort-Protokoll. Der OCSP Kunde sendet eine OCSP-Anfrage an den OCSP *Responder*, einen Server. Mit der Anfrage bittet der Kunde den Responder um den Status eines oder mehrerer Zertifikate. In der Antwort

gibt der Responder die Zertifikatestatusinformation, wie in Abbildung 121 gezeigt, zurück.

Eine Anfrager erreicht eine Zertifikate ausstellende CA, d.h. einen designierten autorisierten Antwortgeber der ausstellenden CA oder einen Antwortgeber („Responder“), dem der Anfragende hinsichtlich der Korrektheit der Zertifikatestatusinformation vertraut. Wenn dabei das Zertifikat der CA und des designierten OCSP „Responders“ unterschiedlich sind, muss das Zertifikat für den OCSP Responder direkt von der CA ausgestellt sein. So wird garantiert, dass der Teilnehmer dem OCSP Responder trauen kann, wenn er der CA traut. Ein OCSP Responder kann direkt mit der CA verbunden sein und dadurch Statusinformation in Echtzeit haben oder er kann einfach erhaltene Sperrlisten auswerten. Gewöhnlich sollten CAs Referenzen an autorisierte OCSP Responder in die Zertifikate einschließen.

Die Antwort eines OCSP Responder muss entweder von der ausstellende CA oder einer entsprechenden autorisierten Stelle beim OCSP oder von einem vertrauensvollen Responder, dessen Schlüssel der Anfrager traut, zu signieren. Die Antwort enthält unter anderem den Informationsstatus über das angefragte Zerrtifikat aus der Anfrage. Hinsichtlich des Zertifikatestatus enthält der OCSP Standard die folgenden Indikatoren:

- “*Good*”: Dies heißt, der Responder hat keine Information, dass das Zertifikat gesperrt ist. Dies bedeutet nicht notwendigerweise, dass das Zertifikat nicht in Wirklichkeit gesperrt ist
- “*Revoked*”: Dies heißt, das Zertifikat ist definitiv gesperrt.
- “*Unknown*”: Dies heiß, der Responder kann keine Angabe hinsichtlich des Zertifikats machen.

Neben diesen Ergebnissen kann eine OCSP Antwort verschiedene Informationen enthalten, wie die Gültigkeitsdauer des Zertifikatestatus und der OSCP Antwort. Es gibt mehrere Semantiken, die von einem OCSP Responder benutzt werden können. Der Zeitpunkt wird angegeben, zu dem der Status in der Antwort bekanntermaßen korrekt war. Ein anderer Zeiteintrag gibt an, wann die nächste Information über den Status erhaltbar ist. Ferner kann eine OCSP Antwort den Zeitpunkt enthalten, an dem der OCSP Responder die Antwort signiert hat. Es ist auch möglich, dass der OCSP Responder signierte OCSP Antworten vorproduziert. Die vorproduzierten Antworten spezifizieren den Status eines Zertifikates zu einer bestimmten Zeit.

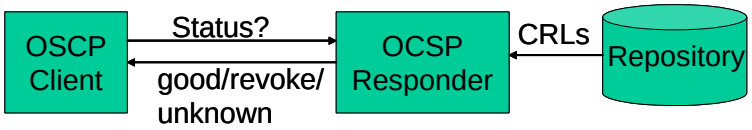


Abbildung 121: OCSP Protokoll

6.3 PKI für das RSP Geschäftsmodell

In diesem Kapitel wird ein spezieller PKI Rahmen beschrieben, der die Grundlage für die weiter unten gezeigte Lösung ist. In diesem Zusammenhang wird gezeigt, wie spezielle PKI assoziierte Funktionen mit den Parteien des RSP-Modells verknüpft werden. Danach wird gezeigt, wie die CAs unterschiedlicher RSP Domänen einander über ein Kreuz-Zertifikat zertifizieren. Weiter wird erklärt, wie Unternehmen-PKIs die Authentifizierung mobiler Nutzer damit durchführen. Es wird erklärt, wo PKI Server und die korrespondierenden Client, die den Zertifizierungsprozess unterstützen, zu positionieren sind. Das Ziel ist es, einen geringen Aufwand für die Zertifikatepfadvalidierung und schließlich für die auf Zertifikaten basierende Authentifikation durch eine obere Grenze für die Pfadlänge zu garantieren.

Im RSP Modell ist zwischen einem gewöhnlichen Teilnehmer und einem Teilnehmer, der Mitarbeiter eines Unternehmens ist, zu unterscheiden. Im Falle des gewöhnlichen Teilnehmers liegt die Funktionalität der RA beim ISP, und die Funktionalität der CA ist an den RSP gebunden. Die CA kann dabei vom RSP selbst oder bei einer dritten Partei betrieben werden, die ihren CA Dienst dem RSP anbietet. Der Einfachheit halber wird im folgenden angenommen, dass die CA vom RSP selbst betrieben wird. Die Allgemeingültigkeit der Lösung hier wird durch diese Annahme nicht eingeschränkt. In der Registrierungsphase sendet der Kunde seine persönlichen Informationen zur Registrierung an den ISP. Dann verifiziert der ISP diese Daten, erzeugt eine Zertifizierungsanfrage und sendet diese an den RSP. Der RSP generiert das Zertifikat und leitet dieses direkt oder über den ISP an den Kunden. Dieser Prozess sowie die Zuordnung der PKI Funktionalität zu den Rollen im RSP Modell sind in Abbildung 122 dargestellt.

Die RSP CA liefert auch die Stellen mit der Statusinformation hinsichtlich der Zertifikate, welche von der CA ausgestellt werden. Wenn ein Nutzerzertifikat gesperrt ist, ist dies von der RSP CA veranlasst worden. Nach der Sperrung eines Nutzerzertifikates macht die RSP CA diese Information den anderen zugänglich. Es ist Sache des RSP festzulegen, welche Teilnehmer diese Sperrinformation erhalten. Die Sperre kann vom Nutzer selbst initiiert werden, z. B. im Falle eines Vergleichs von Schlüsseln des Nutzers oder vom ISP, wenn der Kontakt zwischen dem ISP und dem Nutzer unterbrochen wird, da der Nutzer die Erlaubnis verloren hat, die ISP Dienste aufzurufen.

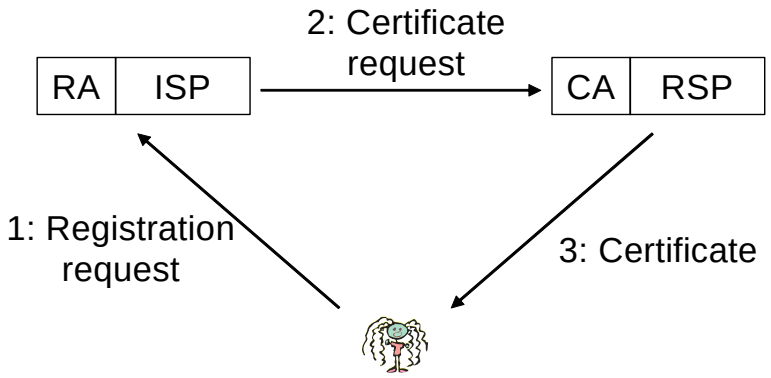


Abbildung 122: Zuordnung der PKI Funktionalität im Falle eines gewöhnlichen Nutzers

Im Falle eines Unternehmensteilnehmers hat man es mit zwei CAs und zwei RAs zu tun. Ähnlich wie im vorigen Fall liegt eine RA beim ISP und eine CA beim RSP. Weiter bearbeitet das Unternehmen selbst die PKI Komponenten oder benutzt den PKI Dienst einer dritten Partei, so dass eine Unternehmens-RA und eine Unternehmens-CA dazwischen geschaltet sind. Der Einfachheit halber wird wiederum angenommen, dass ohne Verletzung der Allgemeingültigkeit kein externer PKI Dienst benutzt wird. Wenn das Unternehmen beim ISP um die Nutzung eines ISP Dienstes anfragt, sorgt der ISP dafür, dass die Unternehmens-CA von der CA des RSP zertifiziert wird. Die Zertifizierung der Unternehmens-CA von der RSP-CA kann als Einführung einer zusätzlichen Hierarchieebene angesehen werden oder als eine uni-direktionalen Über-Kreuz-Zertifizierung. Angestellte des Unternehmens erhalten ihr Unternehmenszertifikat von der Unternehmens-CA. Der ganze Unternehmenszertifizierungsprozess wird initiiert, wenn ein Angestellter ein Unternehmenszertifikat anfordert. Um ein Zertifikat zu erhalten, sendet der Angestellte seine Daten an eine interne RA oder dies wird automatisch für den Angestellten durch interne administrative Strukturen initiiert. Danach generiert die Unternehmens-CA das Unternehmenszertifikat für den Angestellten und gibt dieses weiter. Die Abbildung 123 zeigt die Verhältnisse.

Beide CAs in Abbildung 123 können Sperrinformationen liefern. Jede CA stellt dabei die Sperrinformation für jene Zertifikate bereit, die sie generiert und herausgegeben hat. Dabei ist die Unternehmens-CA für die Sperrung des Angestelltenzertifikats verantwortlich. Dies ist z. B. wichtig, wenn ein Angestellter das Unternehmen verlässt und nicht länger als Mitarbeiter des Unternehmens erkannt werden soll. Damit wird ein ehemaliger Angestellter von der Nutzung der Unternehmensressourcen und der Verursachung weiterer unerwünschter Kosten für das Unternehmen ausgeschlossen. Die RSP CA ist für die Sperre des Zertifikats der Unternehmens-CA verantwortlich. Dies muss erfolgen, wenn der ISP und das Unternehmen ihre Geschäftsbeziehungen abbrechen. Als Folge sind sofort alle Angestellten von der Nutzung der mobilen Kommunikationsdienste ausgeschlossen, die auf dem entsprechenden Vertragsabschluß beruhen. Dies bedeutet, dass, wenn immer ein Angestellter versucht, einen mobilen Kommunikationsdienst anzusprechen, die Gültigkeit des Zertifikatepfades zu überprüfen ist, was mindestens die Verarbeitung der Statusinformationen beinhaltet, die von der

Unternehmens-CA und von der RSP CA geliefert werden können. Allerdings können die Angestelltenzertifikate und auch die CA-Zertifikate gültig bleiben, soweit sie in anderen Zusammenhängen angewendet werden, die keine Informationssperre der RSP CA erfordern. Z. B. könnte dies die Nutzung von Zertifikaten für interne Zwecke innerhalb eines Unternehmens betreffen.

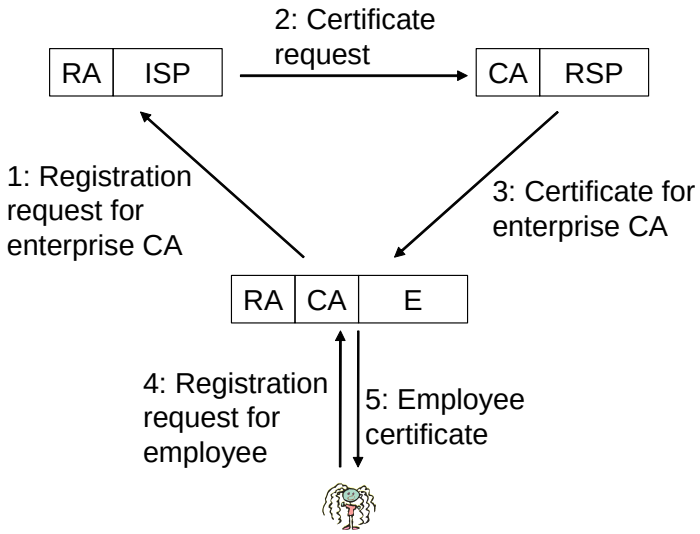


Abbildung 123: Zuordnung der PKI Funktionalität im Fall eines Angestellten als Nutzer

Abbildung 124 stellt die Beziehungen zwischen ISPs, Unternehmen und Nutzern dar und wie sie Zertifizierungsdienste nutzen. Ein RSP kann vertragsgemäß Beziehungen zu mehreren ISPs haben, die wiederum Beziehungen mit mehreren Kunden haben können. Diese Kunden können Unternehmensmitarbeiter oder gewöhnliche Teilnehmer sein. Da alle diese Parteien von den PKI Diensten eines RSP betreut werden, gehören sie zusammen derselben RSP Domäne an. In der Praxis kann man jedoch nicht annehmen, dass nur ein RSP vorhanden ist. Es wird mehrere RSPs geben, die ihre Dienste anbieten und somit auch mehrere RSP Domänen. RSP bieten ihre Dienste entsprechend geographischer Gebietseinteilungen an, z.B. in länder- oder regionspezifischen Domänen oder sie sind Konkurrenten. Für Nutzer ist es sehr wünschenswert, dass sie Dienste von ISPs nutzen können, die zu unterschiedlichen RSP Domänen gehören. Dies erfordert, dass die RSPs so kooperieren, dass Konstruktion und Validierung von Zertifikateketten quer über die RSP Domänengrenzen hinweg gemacht werden können

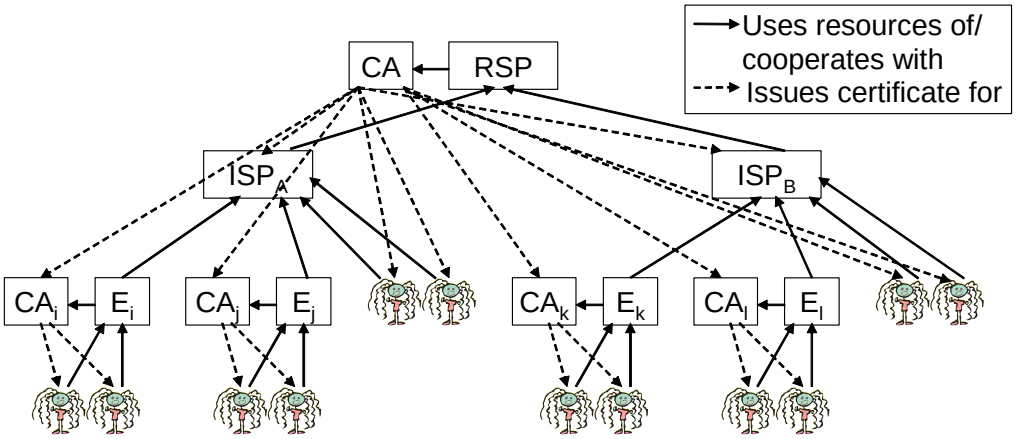


Abbildung 124: ISPs, Unternehmen und Nutzer innerhalb derselben RSP Domäne

Wenn ein mobiler Teilnehmer, der zu einem spezifischen ISP gehört, irgendeinen ISP anspricht, der zu einem anderen ISP gehört, dann müssen spezifische Forderungen von den korrespondierenden ISPs erfüllt werden, d.h. die RSPs müssen gewisse Beziehungen aufbauen. Dazu gibt es im Allgemeinen mehrere Möglichkeiten. Es werden drei Möglichkeiten für die angenommene Kooperation zwischen den RSPs und den mit ihnen korrespondierenden CAs unterstellt:

- Kreuz-Zertifizierung,
- Definition geeigneter Vertrauensanker und
- Keine Kooperation.

Die Kooperation der RSP CAs beruhe auf der Nutzung der Kreuz-Zertifizierung, z.B. kreuz-zertifizieren an RSPs assoziierte CAs einander gegenseitig. Diese Kreuz-Zertifizierung kann uni- oder bidirektional sein. Die Kreuz-Zertifzierung liefert Zertifkationspfade über unterschiedliche RSP-Domänen hinweg. Wenn alle RSPs bidirektional sich gegenseitig kreuz-zertifizieren, dann ist garantiert, dass immer ein Zertifikatepfad zur eigenen RSP Domäne existiert, dass z. B. ein Zertifikatepfad bei der RSP CA der eigenen RSP Domäne endet. Diese Zertifikatepfade beziehen nicht mehr als zwei RSPs ein. Somit kann eine obere Grenze für die Länge der Zertifikatepfade festgelegt werden. Abbildung 125 zeigt die beidseitige Kreuz-Zertifizierung der CAs, die den entsprechenden RSPs assoziiert sind.

Wenn die RSPs ihre Zusammenarbeit beenden wollen, können sie die Kreuz-Zertifikate sperren, und die Verbindung zwischen unterschiedlichen RSP Domänen ist aufgehoben. Das CA Zertifikat, an welches das Kreuz-Zertifikat gebunden ist, bleibt weiterhin gültig

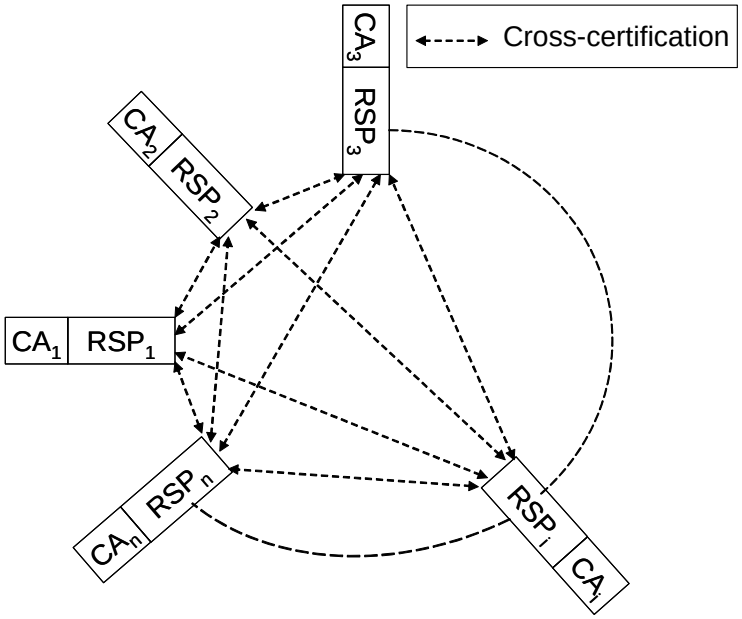


Abbildung 125: Beidseitige Kreuz-Zertifizierung von CAs assoziiert zu RSPs

RSPs können auch beschließen, die einem anderen RSP assoziierte CA als Vertrauensanker zu definieren. Dies hat den Vorteil, dass der Zertifikatepfad im Vergleich zu dem Ansatz der Kreuz-Zertifizierung verkürzt werden kann. In einem Zertifikatepfad ist das letzte zu validierende Glied das Zertifikat, welches von einer einen Vertrauensanker darstellenden CA generiert ist. Für die Verifikation dieses Zertifikates muss ein öffentlicher Schlüssel einer entsprechenden vertrauenswürdigen CA allen jenen RSPs zur Verfügung stehen, die dieser CA vertrauen. Die Aggregation aller CAs, die als Vertrauensanker für die RSP_i dienen, bilden die Menge der Vertrauensanker $TAS_i = \{ca_{i_1}, ca_{i_2}, \ldots, ca_{i_n}\}$ der RSP_i.

RSPs können auch entscheiden, die Zusammenarbeit mit anderen RSPs zu verweigern. Wenn ein RSP nicht mit anderen RSP kooperiert, dann ist die Zertifikate basierte Authentifizierung nur in der RSP eigenen Domäne unterstützt.

Abbildung 126 zeigt unterschiedliche RSP Domänen mit kreuz-zertifizierten RSP CAs. Weiter zeigt die Abbildung die Instanzen, die in Roaming-Szenarien involviert sind, und die Zertifikate dieser Instanzen mit korrespondierenden CAs. Mobile Teilnehmer und ein kontaktierter ISP, der zu einer anderen RSP Domäne gehört, sind zu sehen. Zertifikate und ihre ausstellenden CAs sind farbig gekennzeichnet

Es ist auch möglich, RSP Ketten mit mehr als zwei RSPs zu betrachten. Da es jedoch eines der wichtigen Ziele ist, die für den Authentifizierungsprozess benötigte Zeit möglichst klein zu halten, sollten Zertifizierungspfade über mehr als zwei RSPs

vermieden werden. Auf eine Untersuchung dieser Möglichkeit wird deshalb hier verzichtet.

In der Praxis existieren alle drei Varianten. Es gibt dabei RSPs, die entweder Kreuz-Zertifizierung mit speziellen RSPs oder ihren Einschluss in TAS vorziehen. Dies bedeutet, dass Lösungen mit Kreuz-Zertifizierung und einer geeigneten Definition von TASs nebeneinander existieren. Wenn keine dieser Möglichkeiten von RSPs gewählt ist, kann der mobile Nutzer nicht zu einem ISP gelangen, welcher der entsprechenden RSP Domäne angehört. Jede dieser Varianten erfordert, dass die involvierten RSPs eine spezielle Beziehung zueinander aufgebaut haben, d. h. dass sie sich zumindest gegenseitig kennen. Es gibt jedoch auch RSPs, die keine gegenseitige Beziehung auf der Grundlage der Kreuz-Zertifizierung oder der Vertrauensanker einrichten wollen mit der Folge, dass für deren Nutzer kein Roaming möglich ist. Im Allgemeinen ist es ein Vorteil für RSPs, mit anderen RSPs zu kooperieren, denn ISPs sind sehr daran interessiert, einen RSP aussuchen zu können, der mit vielen anderen RSPs kooperiert, da dann die Dienste des ISPs attraktiver für Nutzer sind.

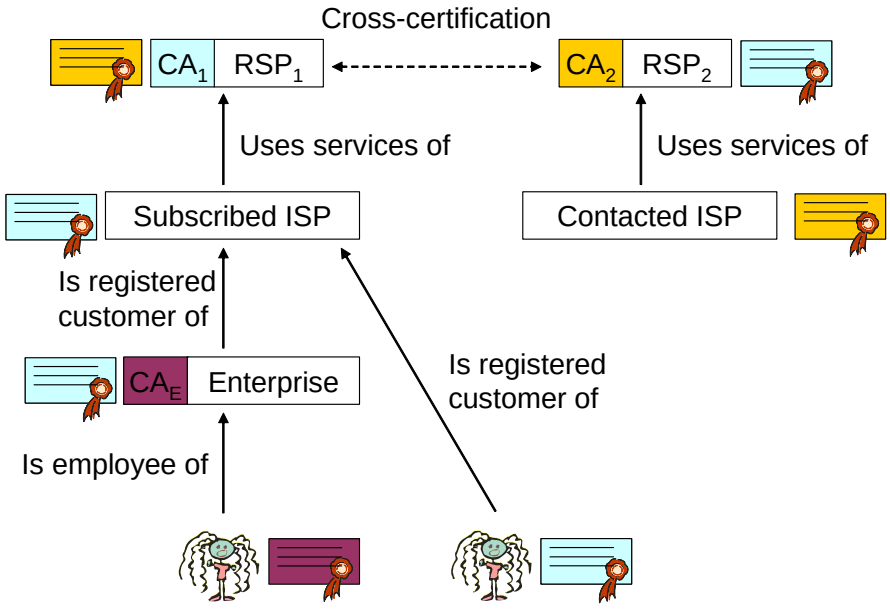


Abbildung 126: RSP Domänen mit kreuz-zertifizierten RSP CAs

Ein weiterer Aspekt ist die Nutzung von PKI Servern (PKIS). Die PKIS sind wichtige PKI Komponenten, die ihre speziellen Dienste mobilen Nutzern, kontaktierten ISPs und grundsätzlich auch anderen PKISs anbieten. Die Nutzung von PKIS erfordert, dass diese mobilen Nutzern, kontaktierten ISPs und anderen PKISs mit den entsprechenden PKI Client ausgerüstet sind. Im hier gemachten Ansatz betreibt jeder RSP einen eigenen PKIS. Ein Unternehmens-CA kann auch seinen eigenen PKIS betreiben. Anstatt eines PKIS kann ein Unternehmen auch CRL-Informationen als einfache CRLs oder als OCSP Dienst bieten. Der Ansatz in dieser Arbeit ist in Abbildung 127 dargestellt. Dort gehört das Unternehmen zur Domäne von RSP_1. Die PKIS werden von den RSP oder den

Unternehmen beliefert und erlauben es anderen Instanzen wie mobilen Nutzern oder kontaktierten ISP, Arbeit an sie zu delegieren. Die spezielle an die PKIS delegierte Arbeit kann unterschiedlich sein. Sie kann von der ausschließlichen Zertifikatepfadkonstruktion über die ausschließliche Erzeugung von Sperrinformationen bis zur Kombination von Zertifikatepfadkonstruktion, von Verifikation der Zertifikatekorrektheiten und von Zertifikatestatusprüfungen reichen. Dies ist besonders von Vorteil, wenn mobile Geräte eine beschränkte Kapazität haben oder wenn ISPs Daten von Zertifizierungspfad Verifikationen nutzen, die bei einem PKIS gespeichert sind.

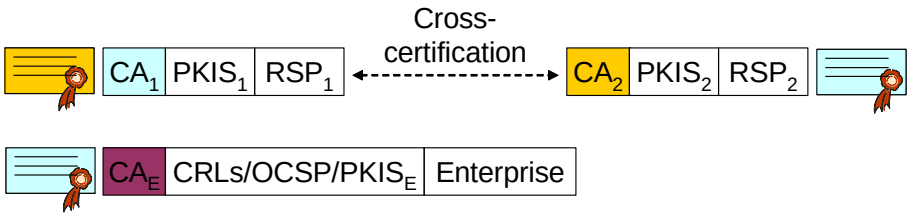


Abbildung 127: PKI Server im RSP Modell

PKISs können in zwei unterschiedlichen Varianten benutzt werden. Einmal führt ein PKIS, an den Arbeit delegiert ist, diese insgesamt selbst aus. Zum anderen kann der PKIS die an ihn delegierte Arbeit an einen weiteren PKIS weiterdelegieren. In manchen Fällen hat die zweite Variante Vorteile. Unternehmen ziehen es möglicherweise vor, ihre CRL Informationen gegenüber der Außenwelt geheim zu halten und den Zugang zu dieser Information nur ausgesuchten Parteien zu gestatten. So kann ein Unternehmen z.B. daran interessiert sein, dass nur ein RSP Zugang zu seinen CRL Daten hat. In der Praxis wird dies der RSP sein, zu dessen Domäne das Unternehmen gehört. Anderen RSPs kann der Zugang zu den Sperrdaten des Unternehmens verwehrt werden. Wenn eine Instanz die Konstruktion und Verifizierung einer Zertifikatekette für einen Mitarbeiter anfordert, der zu einem anderen RSP gehört, kann der PKIS, der von dieser Instanz kontaktiert ist, die delegierte Arbeit selbst nicht ausführen. In diesem Fall delegiert der kontaktierte PKIS diese Arbeit an einen PKIS eines anderen RSPs, der die Sperrlistendaten des Unternehmens einsehen kann.

SCVP ist eine mögliche Technologie für PKIS. Ob jedoch SCVP sich langfristig gegenüber anderen möglichen Technologien durchsetzen wird, ist zurzeit nicht vorhersehbar. PKIS nutzen die PKI Technologie mit Zertifikaten gemäß RFC 3280. PKIS können jedoch auch andere Technologien integrieren und die Sperrung von Zertifikaten und Statusinformationen bearbeiten, wie z.B. CRLs oder OCSP.

6.4 Zertifikate Validierung mobiler Nutzer bei der Authentifikation

Im hier vorliegenden Modell beruht die Nutzerauthentifikation auf Zertifikaten. Wenn Dienste von einem kontaktierten ISP angefordert werden, wird eine gegenseitige

Authentifikation durchgeführt, welche beiden Parteien die Identität des anderen bestätigt. Um Sicherheit zu gewähren, muss die Gültigkeit der Zertifikate jedesmal vom kontaktierten ISP überprüft werden, wenn der Nutzer eine Authentifizierung durchführt.

Es wird zwischen zwei Hauptszenarien für den Ablauf der Nutzerauthentifikation unterschieden. Im ersten Szenario wird ein einzelner RSP betrachtet. Hierbei sind der Nutzer, sein Partner-ISP und sein kontaktierter ISP, alle an denselben RSP angeschlossen. Im zweiten Szenario werden mehrere RSPs betrachtet, wobei ein Nutzer, der von einem kontaktierten RSP authentifiziert werden soll, zu einem ISP in einer entfernten RSP Domäne gehört. Hierbei sind zwei Unterfälle zu betrachten: Ein Nutzer kann entweder ein ISP Kunde oder ein Unternehmensangestellter sein. Im Falle des Unternehmens-angestellten kann er ein Zertifikat besitzen, das von der CA des eigenen Unternehmens ausgestellt ist. Im Folgenden wird jeder Fall behandelt, und weiter werden mögliche Lösungen für die schnelle Nutzerauthentifikation beschrieben. Der Verkehr zwischen den einzelnen Instanzen kann mit unterschiedlichen Protokollen durchgeführt werden wie mit Anfrage/Antwort-OCSP und –SCVP. Die Beschreibung ist jedoch allgemein gehalten ohne spezielle Protokolle. Das jeweilige Protokoll kann dann entsprechend der verfügbaren Technik ausgewählt werden.

6.4.1 Mobiler Nutzer und kontaktierter ISP in derselben RSP-Domäne

In diesem Abschnitt wird der Fall betrachtet, in dem der mobile Nutzer und der kontaktierte ISP sich in derselben RSP Domäne befinden. Dabei sind die Unterfälle eines Nutzers, der Mitarbeiter eines Unternehmens ist, und eines „normalen“ Nutzers zu betrachten.

6.4.1.1 Normaler Nutzer ohne Unternehmen

Ein Nutzer, der vom kontaktierten ISP authentifiziert werden soll, ist Kunde eines ISPs, welcher zur selben RSP-Domäne gehört. Weiter ist die CA, welche das Nutzerzertifikat liefert, diesem RSP assoziiert.

Im Authentifizierungsprotokoll versorgt der Nutzer den kontaktierten ISP mit seinem Zertifikat. Da das Nutzerzertifikat von derselben CA herausgegeben ist, wie das Zertifikat des ISP, kann angenommen werden, dass der kontaktierte ISP selbst das Nutzerzertifikat validieren und seinen Status überprüfen kann. Er kann diese Arbeit jedoch auch an einen PKIS delegieren.

Im ersten Fall delegiere der kontaktierte ISP diese Arbeit nicht an einen PKLS. Wenn der kontaktierte ISP das Nutzerzertifikat erhält, extrahiert er die Information über den Aussteller aus dem Zertifikat. Da es derselbe Aussteller, wie der seines eigenen Zertifikates ist, hat er den öffentlichen Schlüssel des Ausstellers verfügbar. Weiter wird angenommen, dass die RSP CA ein Vertrauensanker TA des kontaktierten ISPs ist. Somit kann er selbst die Richtigkeit des Nutzerzertifikates mit dem öffentlichen Schlüssel des Ausstellers verifizieren. Nach der Verifikation sendet der kontaktierte ISP eine Anfrage an die Aussteller CA nach der neuesten Sperrinformation. Die Aussteller CA antwortet, ob das Zertifikat gesperrt ist oder nicht. Der kontaktierte ISP wertet die Antwort über die

Sperrinformation aus und fährt mit dem Nutzerauthentifikationsprotokoll fort. Diese Zusammenarbeit zwischen mobilem Nutzer, kontaktiertem ISP und RSP ist in Abbildung 128 dargestellt.

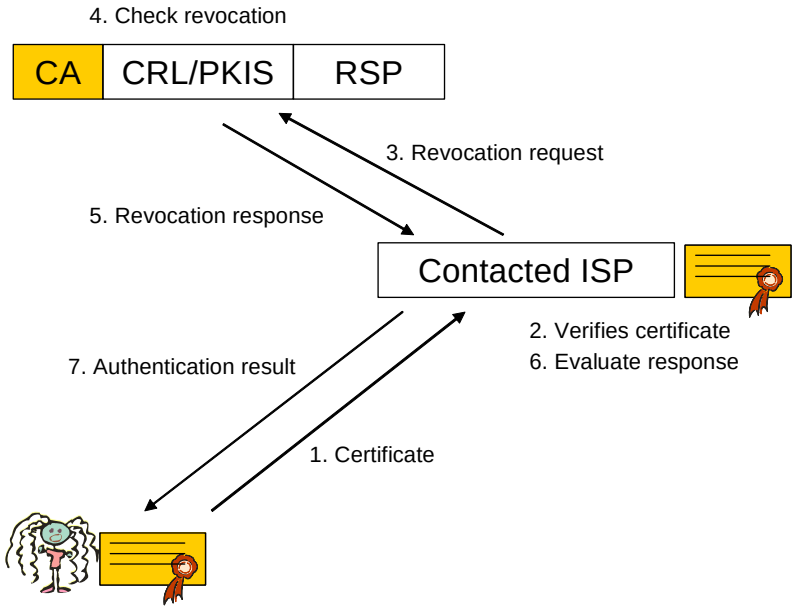


Abbildung 128: Authentifizierung normaler Nutzer ohne Delegation zum PKIS

Jetzt wird der Fall betrachtet, in dem die Zertifikateverifikation vollständig an einen PKIS delegiert wird. Dieser Fall ist in Abbildung 129 dargestellt. Hier erzeugt der kontaktierte ISP eine Anfrage für die Verifikation des Nutzerzertifikats, nachdem er das Zertifikat vom Nutzer erhalten hat. Er sendet diese Anfrage an den PKIS, der von seinem RSP bereitgestellt ist, um das Zertifikat zu verifizieren und den Status des Zertifikats zu prüfen. Der PKIS ist immer in der Lage, das Zertifikat zu verifizieren, da der Aussteller CA mit ihm assoziiert ist und somit in der Menge der Vertrauensanker TAS des PKIS enthalten ist. Der PKIS des RSP verifiziert die Korrektheit des Nutzerzertifikates. Dann prüft er die Sperrinformation und gibt das Ergebnis an den kontaktierten ISP zurück.

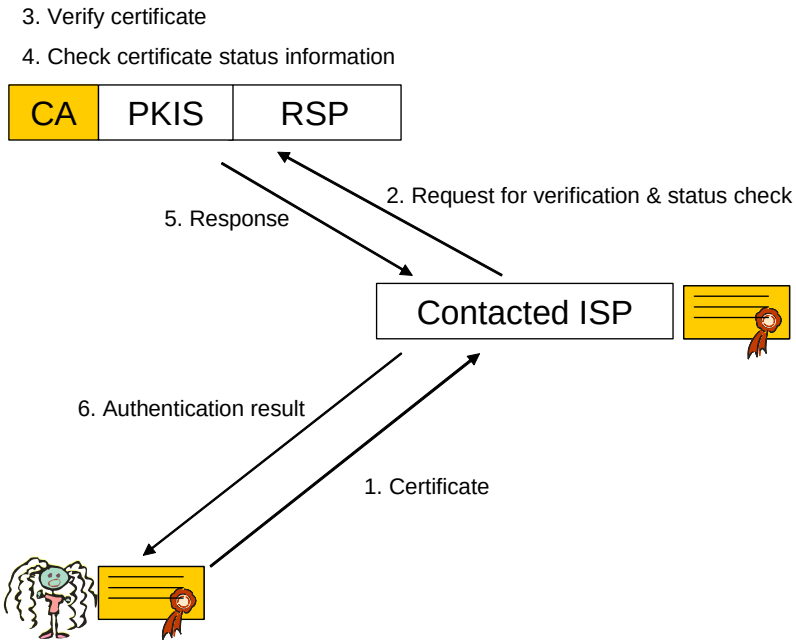


Abbildung 129: Authentifizierung normaler Nutzer mit Delegation zum PKIS

Abhängig von dieser Antwort bearbeitet der kontaktierte ISP dann das Authentifikationsprotokoll weiter.

6.4.1.2 Unternehmensmitarbeiter als Nutzer

In diesem Fall ist der Nutzer, der vom kontaktierten ISP authentifiziert werden soll, ein Mitarbeiter eines Unternehmens in derselben RSP-Domäne. Er erhält sein Zertifikat von der CA seines Unternehmens.

Es wird nun angenommen, dass der kontaktierte ISP die Arbeit für die Zertifikateverifizierung und für die Überprüfung des Zertifikatestatus an den PKIS delegiert, welcher für den RSP arbeitet. Die ausstellende CA des Nutzerzertifikates muss dabei dem kontaktierten ISP nicht bekannt sein. Deshalb sendet der kontaktierte ISP eine Anfrage an den PKIS zur Verifizierung des Zertifikates und der zugehörigen Zertifikatekette. Darauf entnimmt der PKIS die Information über die Aussteller CA aus dem Zertifikat und sendet eine Anfrage an die Unternehmens CA (CA_E), um die Zertifikate desselben zu erhalten. Z. B. können diese die selbst signierten Zertifikate der CA_E und ein Kreuz-Zertifikat auf dem selbstsignierten von der RSP CA ausgestellten Zertifikat enthalten. Nachdem die Unternehmens CA alle ihre Zertifikate zurückgesandt hat, kann der PKIS einen Validationspfad erzeugen, in dem die RSP CA als Vertrauensanker dient[25]. Der PKIS verifiziert dann die Zertifikatekette mit dem Zertifikat des mobilen Nutzers, mit dem der Unternehmens CA und dem vom RSP CA ausgestellten Kreuz-Zertifikat mit den öffentlichen Schlüsseln. Dann prüft der PKIS den Status aller dieser Zertifikate. Wenn dies alles erledigt ist, sendet der PKIS das Ergebnis

[25] Die RSP CA ist in der Vertrauensankerkette des von demselben RSP betriebenen PKIS enthalten.

zurück an den kontaktierten ISP, welcher sich entsprechend dem Ergebnis verhält. Die Interaktionen sind in Abbildung 130 dargestellt.

6.4.2 Nutzer und kontaktierter ISP in unterschiedlichen RSP Domänen

In diesen Abschnitt wird der Fall der zertifikatebasierten Nutzerauthentifikation behandelt, bei dem der mobile Nutzer und der kontaktierte ISP zu unterschiedlichen RSP Domänen gehören. Es wird wieder zwischen normalem Nutzer und Unternehmensmitarbeiter unterschieden. Weiter kann unterschieden werden, wie diese Fälle von der Kooperation der involvierten RSPs abhängen und wie ihre PKIS zusammenarbeiten. Im Weiteren werden bei der Behandlung von unterschiedlichen RSP Domänen der dem mobilen Nutzer (Unternehmensmitarbeiter oder nicht) assoziierte RSP als RSP_1 und der dem kontaktierten ISP assoziierte RSP als RSP_2 bezeichnet.

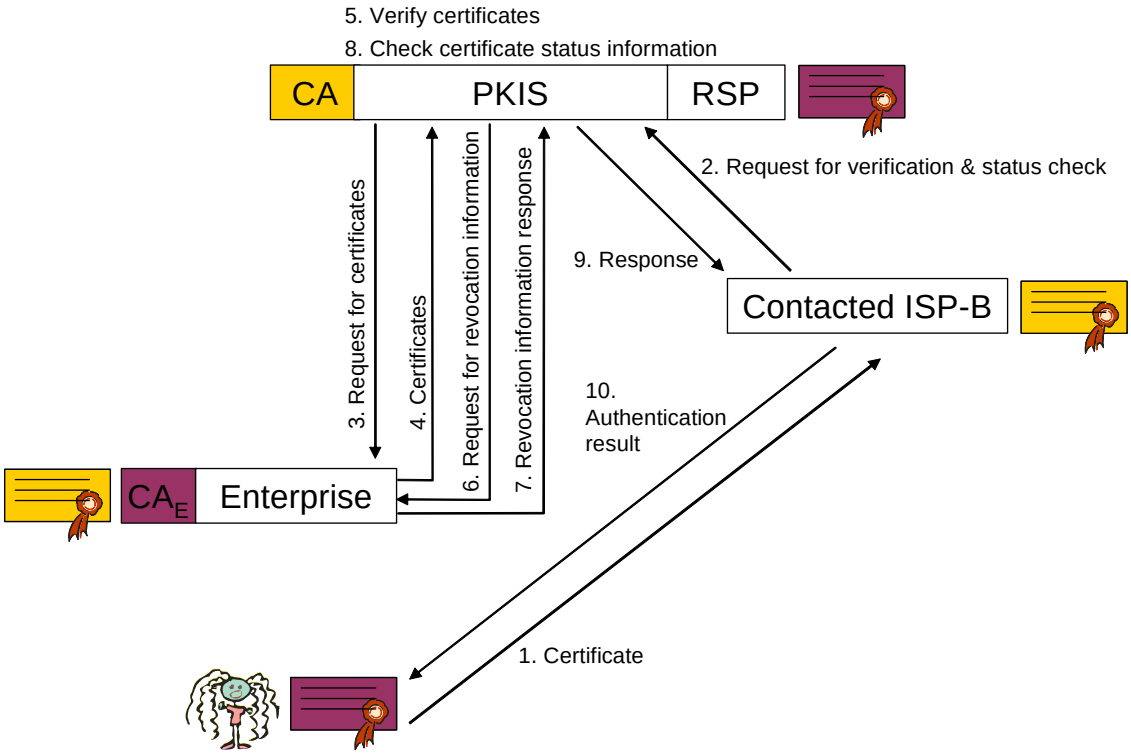


Abbildung 130: Nutzerauthentifizierung für Unternehmensmitarbeiter mit Delegation zum PKIS

6.4.2.1 Der normale Nutzer ohne Unternehmen

Im Folgenden wird der normale Nutzer behandelt, der z.B. sein Zertifikat von der CA erhält, die von RSP_1 betrieben wird. Drei Unterfälle werden betrachtet:

- Kreuz-zertifizierte RSPs,
- RSPs als Vertrauensanker,
- Re-Delegation an andere PKISs.

6.4.2.1.1 Kreuz-zertifizierte RSPs

In diesem Fall ist ein Nutzer, der vom kontaktierten ISP authentifiziert werden soll, Kunde eines ISP einer anderen RSP Domäne. Es wird angenommen, dass die RSP CAs, CA_1 und CA_2, die mit diesen Domänen assoziiert sind, sich gegenseitig kreuz-zertifiziert haben. Soll ein Nutzer vom kontaktierten ISP authentifiziert werden, delegiert der kontaktierte ISP die Arbeit der Konstruktion des Zertifikatepfades, die Verifikation der Zertifikategültigkeit und der Prüfung des Sperrzustandes an $PKIS_2$, der von RSP_2 betrieben wird. CA_1 sei in der Menge der Vertrauensanker von $PKIS_2$ nicht enthalten, so dass $PKIS_2$ versucht, einen Zertifikatepfad mit dem Vertrauensanker CA_2 zu erstellen

Nachdem die Aufgabe an $PKIS_2$ delegiert worden ist, baut $PKIS_2$ einen Zertifikatepfad auf, der das Zertifikat des mobilen Nutzers, das Zertifikat von CA_1, was eventuell selbstsigniert ist, und das Kreuz-Zertifikat von CA_1, das von CA_2 erzeugt und ein Vertrauensanker von $PKIS_2$ ist, enthält. Nach der Erzeugung dieses Pfades wird überprüft, ob die Zertifikate korrekt sind. Wenn alle Resultate dies bestätigen, dann initiiert $PKIS_2$ die Prüfung des Zertifikatestatus, d.h. die Prüfung des Status des Nutzerzertifikates durch Informationen, die von CA_1 bereitgestellt sind, und die Prüfung des Kreuz-Zertifikates von CA_1, das von CA_2 bereitgestellt ist. Die Prüfung des Status eines selbst signierten Zertifikates macht allerdings keinen Sinn.

Der Ablauf ist im Detail folgender: Wenn der Nutzer sein Zertifikat an den kontaktierten ISP gesendet hat, sendet dieser dieses Zertifikat in einem Delegationsauftrag an den $PKIS_2$. Der $PKIS_2$ extrahiert dann die Information hinsichtlich des Zertifikatausstellers aus dem Zertifikat und fordert von CA_1 seine Zertifikate an. Wenn CA_1 mit einem selbst signierten Zertifikat und dem von CA_2 erzeugten Kreuzzertifikat antwortet, ist ein geeigneter Zertifikatepfad konstruiert, die Korrektheit der Zertifikate ist verifiziert und der Sperrstatus ist überprüft. Dabei werden Sperrinformationen von CA_1 und CA_2 verlangt. Die Beziehungen sind im Detail in Abbildung 131 zu ersehen.

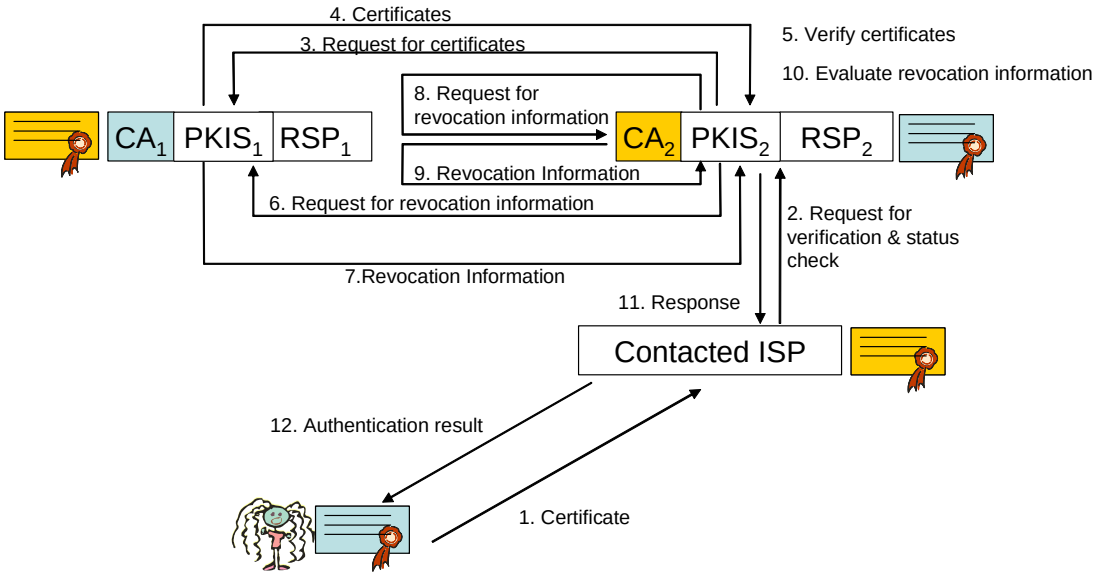


Abbildung 131: Nutzerauthentifizierung mit kreuz-zertifizierten RSPs aus unterschiedlichen Domänen

Schlussendlich wird das Ergebnis von $PKIS_2$ zum kontaktierten ISP zurückgesandt, der sich dann entsprechend den Verifikationsergebnissen verhält.

6.4.2.1.2 RSPs als Vertrauensanker

Die gleichen Annahmen wie im vorigen Fall werden hier zu Grunde gelegt. Der Nutzer, der vom kontaktierten ISP authentifiziert werden soll, ist Kunde eines ISPs einer anderen RSP Domäne. Jedoch sei CA_1 jetzt in der Menge der Vertrauensanker von CA_2 enthalten, d.h. Zertifikatspfade müssen nur soweit validiert werden; bis der Pfad CA_1 erreicht. Dies ist ein Vorteil für den Aufwand für die Konstruktion des Zertifikatepfades, für die Verifikation der Zertifikatekorrektheit und für die Zertifikateüberprüfung verglichen mit der vorangegangenen Lösung. Diese Lösung hat eine positive Wirkung hinsichtlich einer schnellen Authentifizierung.

Wenn der kontaktierte ISP die Arbeit an den $PKIS_2$ delegiert, bemerkt der $PKIS_2$ sofort, dass das Zertifikat von einem seiner Vertrauensanker erzeugt wurde. Als Folge ist die Konstruktion des Zertifikatepfades an dieser Stelle fertig. Da außerdem der $PKIS_2$ den öffentlichen Schlüssel von CA_1 zur Verfügung hat, muss dieser nicht nochmals angefordert werden. Wenn der öffentliche Schlüssel erkannt ist, kann $PKIS_2$ nun das Nutzerzertifikat validieren. Zur Überprüfung des Status des Zertifikates fragt $PKIS_2$ die CA_1 nach Sperrinformationen. Danach versorgt $PKIS_2$ den kontaktierten ISP mit dem entsprechenden Ergebnis, gemäß dem die weitere Interaktion zwischen dem kontaktierten ISP und dem mobilen Nutzer abläuft. Der gesamte Prozess ist in Abbildung 132 dargestellt.

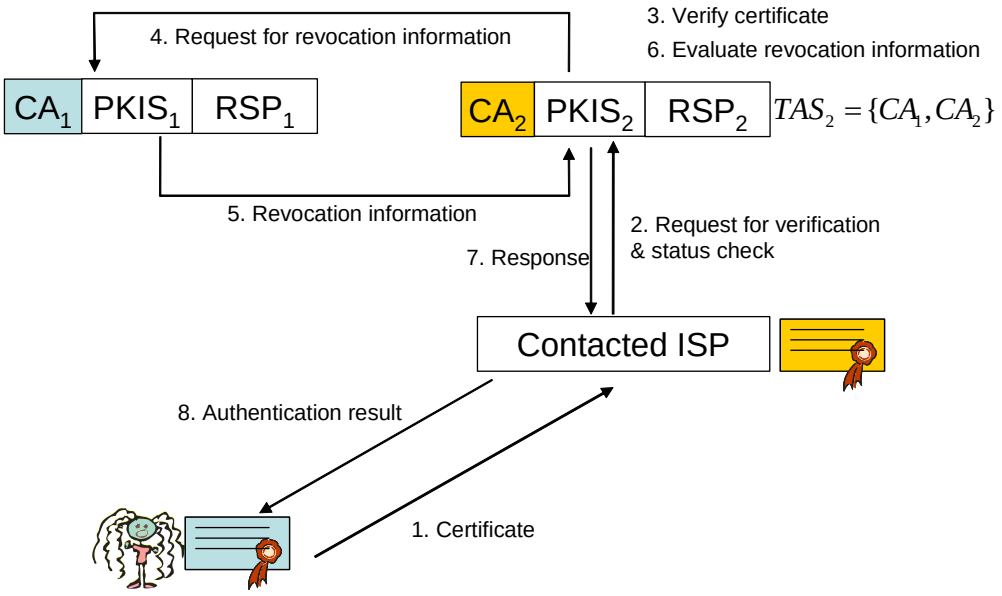


Abbildung 132: Nutzerauthentifizierung mit fremden RSPs als Vertrauensanker

6.4.2.1.3 Re-Delegation an andere PKISs

Die Bedingungen sind wieder dieselben wie in den vorangegangenen Fällen. Der Unterschied liegt darin, dass der $PKIS_2$ seine Arbeit an den PKIS des korrespondierenden RSP zurückdelegiert, d.h. an den $PKIS_1$. Somit ist für andere Parteien keine Notwendigkeit vorhanden, auf CA_1 zuzugreifen. Dies hat den Vorteil, dass eine CA wie CA_1 keinen Zugriff auf ihre Sperrdaten für andere vorsehen muss. ISPs sollten die Daten ihrer Kunden geheim halten. Dieser Ansatz ist ein Beitrag zum Ziel der Wahrung der Privatsphäre.

Ein anderer Vorteil dieses Ansatzes ist die Reduktion der Länge des Zertifikatepfades verglichen mit dem Ansatz der Kreuz-Zertifzierung. Dies liegt darin, dass $PKIS_1$, an den die Arbeit weiterdelegiert ist, nur einen Zertifikatepfad der Länge eins zu verarbeiten hat. Bevor $PKIS_2$ die Arbeit an $PKIS_1$ delegieren kann, muss $PKIS_2$ wissen, an welche Adresse dies erfolgen kann. Deshalb wird eine Zertifikateerweiterung vorgesehen, welche die Adresse anzeigt, an die die Arbeit delegiert werden kann. Grundsätzlich kann die Ausstelleradresse von CA_1 als Referenz dienen für die PKIS Adresse für die Redelegation. Für einen Unternehmensangestellten ist dies jedoch nicht immer möglich, wie dies in der Beschreibung der Re-Delegation im Falle des Unternehmensmitarbeiters als Nutzer zu sehen ist. Für eine einheitliche Behandlung der Re-Delegation wird eine X.509 Zertifikateerweiterung vorgeschlagen, welche eine Referenz für Re-Delegation vorsieht. Diese X.509 Zertifikateerweiterung wird *RSP PKIS „Extension“* genannt. Damit kann im Allgemeinen ein kontaktierter ISP eine Re-Delegation zu einem

korrespondierenden PKIS versuchen, wenn ein Nutzerzertifikat eine solche Erweiterung besitzt.

Wenn $PKIS_2$ eine Anfrage vom kontaktierten ISP erhält, kann er die Anfrage an die Komponente, die in der Zertifikatserweiterung spezifiziert ist, delegieren. Wenn $PKIS_1$ die Anfrage erhalten hat, bemerkt er, dass CA_1 der Herausgeber ist. Da CA_1 ein Element der Menge der Vertrauensanker von $PKIS_1$ ist, ist keine weitere Arbeit zur Konstruktion des Zertifikatepfades notwendig. Dies heißt, dass $PKIS_1$ nur die Korrektheit des Zertifikates verifizieren muss und den Status zu überprüfen hat. Die entsprechende Information hierfür ist wird von CA_1 vorgehalten. Dann sendet $PKIS_1$ seine Antwort an $PKIS_2$. Auf Grundlage dieser Antwort generiert $PKIS_2$ seine Antwort und sendet diese an den kontaktierten ISP. Die detaillierte Beschreibung ist in Abbildung 133 zu sehen.

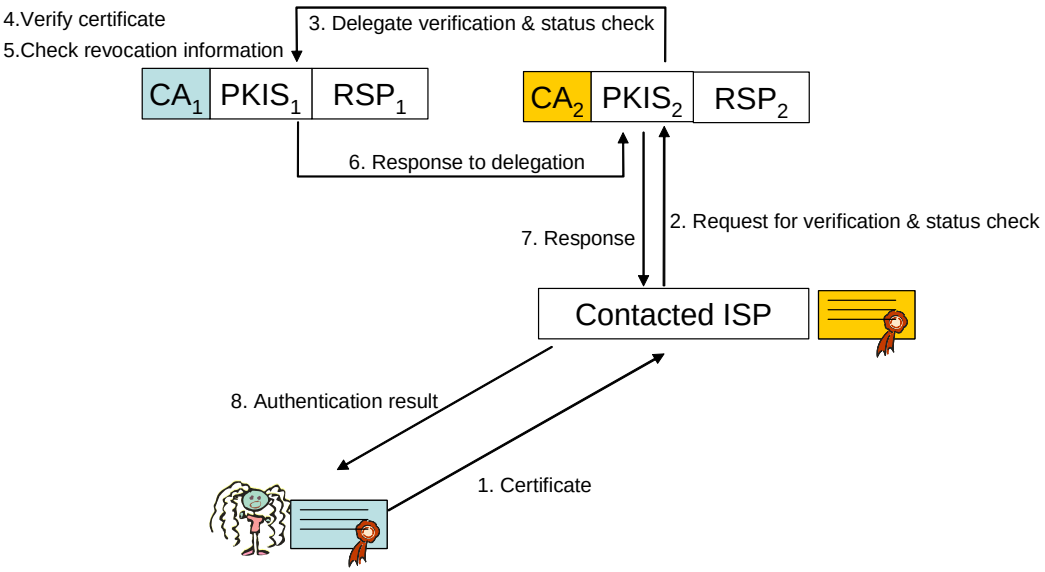


Abbildung 133: Nutzerauthentifizierung mit Re-Delegation zu PKIS anderer RSPs

6.4.2.2 Der Unternehmensangestellte als Nutzer

Im Folgenden wird der Unternehmensangestellte betrachtet, der sein Zertifikat von der CA des Unternehmens erhält. Es werden drei Unterfälle untersucht:

- Kreuz-zertifizierte RSPs,
- Andere RSPs als Vertrauensanker,
- Re-Delegation an andere PKISs.

6.4.2.2.1 Kreuz-zertifizierte RSPs

In diesem Fall ist der Nutzer, der vom kontaktierten ISP authentifiziert werden soll, ein Angestellter eines Unternehmens, das zu einer anderen RSP Domäne gehört. Es wird angenommen, dass die RSP CAs, CA_1 and CA_2, sich kreuz-zertifiziert haben. Weiter ist CA_E von CA_1 kreuz-zertifiziert und CA_E hat das Nutzerzertifikat ausgestellt. Wenn ein Nutzer von einem kontaktierten ISP authentifiziert werden soll, dann delegiert der kontaktierte ISP die Arbeit hinsichtlich der Konstruktion des Zertifizierungspfades, der Verifikation der Gültigkeit der Zertifikate und der Prüfung des Sperrstatus an den $PKIS_2$. Es wird angenommen, dass CA_1 und CA_E nicht in der Menge der Vertrauensanker des $PKIS_2$ enthalten sind, was bedeutet, dass der $PKIS_2$ einen Zertifikatepfad mit dem Vertrauensanker CA_2 zu konstruieren versucht.

Wenn ein Nutzer sich irgendwo anmeldet und versucht, eine Internetverbindung über einen kontaktierten ISP zu bekommen, versorgt er den kontaktierten ISP mit seinem Zertifikat innerhalb des Authentifikationsprotokolls. Dann sendet der kontaktierte ISP eine Anfrage mit dem Nutzerzertifikat an den $PKIS_2$, damit dieser die geforderte Arbeit für ihn erledigt. Der $PKIS_2$ extrahiert die Information über die Aussteller CA des Zertifikates aus diesem, welche in diesem Fall die CA_E ist. Dann fordert er die CA_E auf, ihre Zertifikate zu versenden. Die Antwort der CA_E beinhaltet zwei Zertifikate, das eigene, eventuell selbst signiert, und das Kreuzzertifikat, welches von CA_1 stammt. Entsprechend dieser Antwort fährt der $PKIS_2$ fort, den Zertifikatepfad durch Erfragen der Zertifikate beim CA_1 zu konstruieren. Unter diesen sind möglicherweise ein selbst signiertes Zertifikat und ein Kreuz-Zertifikat, das von CA_2 herausgegeben ist. Dies bedeutet, dass $PKIS_2$ einen seiner Vertrauensanker erreicht hat. Jetzt kann $PKIS_2$ die Korrektheit der Zertifikate verifizieren. Danach beginnt $PKIS_2$, die Sperrinformation für die Zertifikate, die nicht selbst signiert sind, abzurufen. Im einzelnen fragt $PKIS_2$ CA_E nach dem Status des Nutzerzertifikates und dann CA_1 nach dem Status des Kreuzzertifikates von CA_E. Letztlich erhält $PKIS_2$ den Status des Zertifikates von CA_1 von CA_2. Das Ergebnis dieses Verifikationsprozesses wird an den kontaktierten ISP zurückgesandt, der dann bei positiver Antwort den mobilen Nutzer authentifizieren kann. Die gesamten Aktionen zwischen den beteiligten Parteien in Abbildung 134 dargestellt.

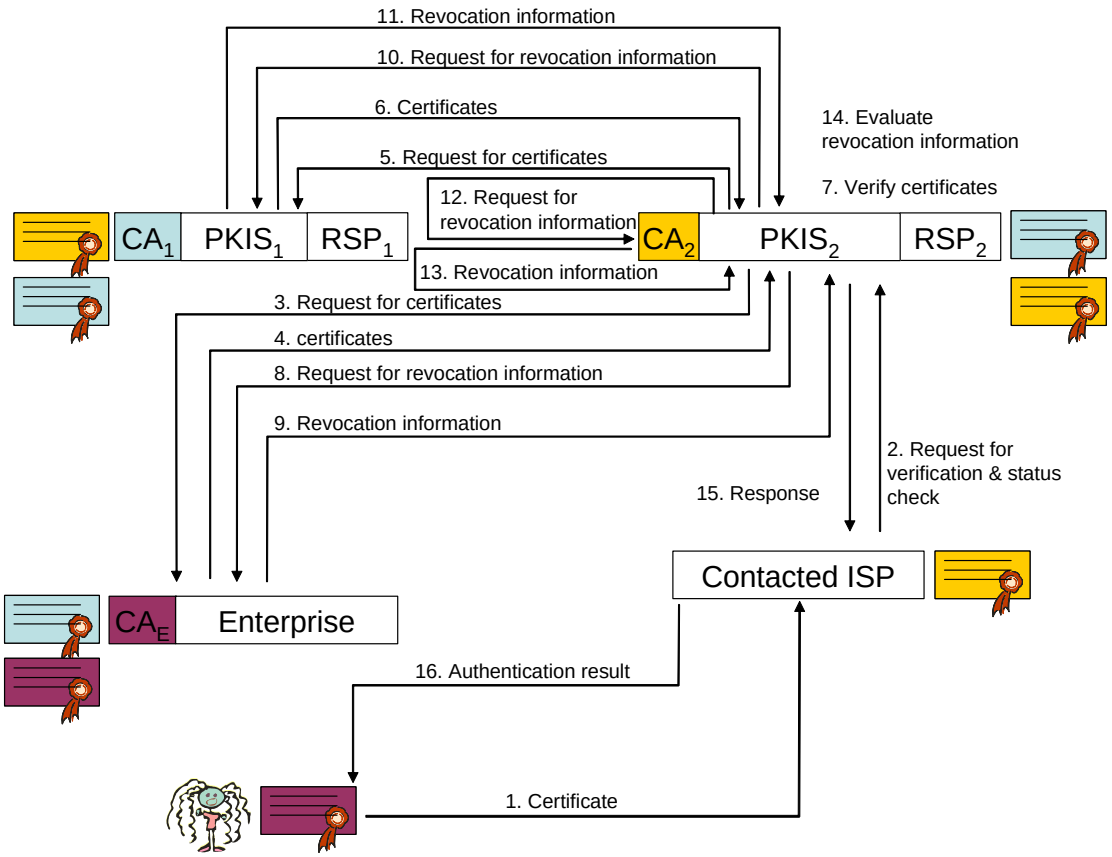


Abbildung 134: Unternehmensnutzer Authentifikation mit kreuz-zertifizierten RSPs in unterschiedlichen Domänen

6.4.2.2.2 Andere RSPs als Vertrauensanker

Die Bedingungen sind ähnlich zu denen des entsprechenden Falles für normale Nutzer. Der Nutzer, welcher vom kontaktierten ISP authentifiziert werden soll, ist ein Angestellter eines Unternehmens, das zu einer anderen RSP Domäne gehört. CA_E hat das Zertifikat des Nutzers ausgestellt, und CA_E ist kreuz-zertifiziert von CA_1. Weiter wird angenommen, dass CA_1 jetzt in der Menge der Vertrauensanker von CA_2 enthalten ist, und dass CA_2 keine spezielle Beziehung zu CA_E hat. Aufgrund der Vertrauensankerkette von CA_2 müssen jedoch Zertifikatepfade nur bis zum Zertifikat von CA_1 validiert werden. Dies ist ein Vorteil hinsichtlich des notwendigen Aufwandes für die Pfadkonstruktion, für die Verifikation der Zertifikatekorrektheit und für die Überprüfung der Zertifikatestatus, verglichen mit der Lösung, bei der CA_1 und CA_2 Kreuz-Zertifikate austauschen. Diese Lösung hat also eine positive Auswirkung auf die Geschwindigkeit der Authentifizierung verglichen mit vorherigen Lösungen.

Die vollständigen Interaktionen sind in Abbildung 135 gezeigt. Nachdem der mobile Nutzer sein Zertifikat zum kontaktierten ISP innerhalb des Authentifikationsprotokolls gesandt hat, verifiziert dieser das Zertifikat nicht selbst. Er sendet es mit einer Anfrage an

den $PKIS_2$, damit alle Arbeiten für die Verifikation für ihn von $PKIS_2$ durchgeführt werden. In einem ersten Schritt extrahiert $PKIS_2$ die Information über die Aussteller CA aus dem Zertifikat, hier die CA_E. Da CA_E nicht in der Menge seiner Vertrauensanker enthalten ist, muss $PKIS_2$ mit der Konstruktion des Zertifikatepfades fortfahren. $PKIS_2$ sendet deshalb eine Anfrage an die CA_E, um deren Zertifikate zu erhalten. CA_E antwortet mit ihrem öffentlichen, eventuell selbst signierten Schlüsselzertifikat und mit ihrem von CA_1 ausgestellten Kreuzzertifikat. Mit diesem Kreuz-Zertifikat hat $PKIS_2$ einen seiner Vertrauensanker erhalten und die Zertifikatepfadkonstruktion ist beendet. Danach kann $PKIS_2$ die Korrektheit der Zertifikate verifizieren. Nachfolgend fordert $PKIS_2$ die Sperrinformation über das Zertifikat des Unternehmensangestellten von CA_E und über das Kreuzzertifikat der CA_E von CA_1 an.

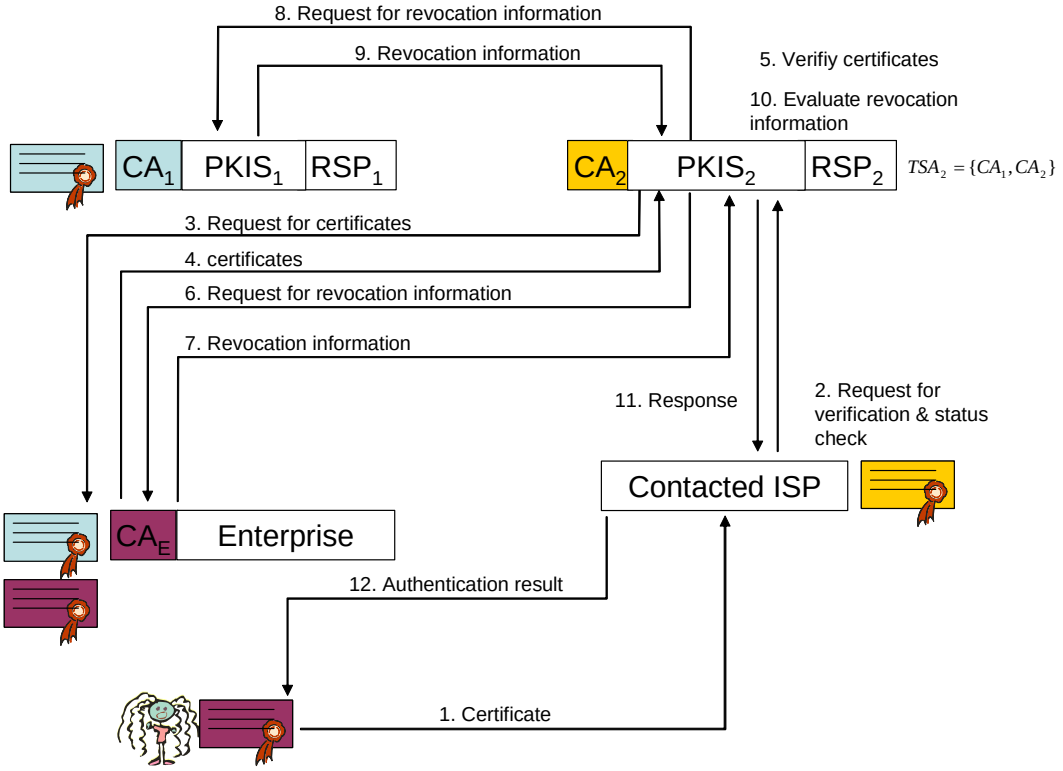


Abbildung 135: Unternehmensnutzerauthentifizierung mit fremden RSPs als Vertrauensanker

Auf diese Weise ist der Prozess der Verifikation des Nutzerzertifikates abgearbeitet und das Ergebnis wird an den kontaktierten ISP gesandt.

6.4.2.2.3 Re-Delegation an andere PKISs

Ähnlich wie im Fall des normalen Teilnehmers delegiert der vom kontaktierten ISP kontaktierte PKIS die Verifikationsarbeit an einen PKIS des entsprechenden RSPs weiter. Dies bedeutet, dass $PKIS_2$ seine Arbeit an den $PKIS_1$ delegiert. Weiter ist das Zertifikat des mobilen Nutzers von CA_E herausgegeben. Dieser Ansatz unterstützt die Geheimhaltung von Informationen im Speicher von CA_E. Mit Ausnahme von CA_1 ist es für die anderen Parteien nicht notwendig, auf den Speicher von CA_E zuzugreifen. Dies hat den Vorteil, dass CA_E keinen Zugriff auf seine Sperrdaten an andere außer CA_1 zulassen muss. Möglicherweise hat das Unternehmen ein Interesse daran, seine Mitarbeiter betreffende Daten geheim zu halten. Dieser Ansatz kann als Beitrag zur Erfüllung der Geheimnisforderung für die Daten eines Unternehmens angesehen werden.

Ein anderer Vorteil dieses Ansatzes ist die Reduktion der Länge des Zertifikatepfades im Vergleich zum Ansatz mit den Kreuz-Zertifikaten. Wie oben schon erwähnt, muss $PKIS_2$ im Falle des normalen Teilnehmers zur Re-Delegation an $PKIS_1$ wissen, an welche Adresse die Arbeit weitergegeben werden kann. Es wurde deshalb eine Erweiterung des Zertifikates vorgeschlagen, die die Adresse enthält, an die die Arbeit weiterdelegiert werden kann. In diesem Fall kann die Erstelleradresse des Zertifikates des Mitarbeiters nicht als Referenz hinsichtlich der PKIS Adresse zur Re-Delegation dienen. Deshalb wird auch hier eine X.509 Zertifikate Erweiterung vorgeschlagen, die eine Referenz für die Re-Delegation aufnimmt. Diese wurde schon im Falle des normalen Teilnehmers vorgeschlagen und *RSP PKIS Extension* genannt.

Die vollständige Lösung für diesen Fall wird in Abbildung 136 gezeigt. Wenn ein kontaktierter ISP innerhalb eines Authentifikationsprotokolls ein Nutzerzertifikat erhält, delegiert er die vollständige Verifikationsarbeit an $PKIS_2$. $PKIS_2$ erkennt die RSP PKIS Zertifikatserweiterung, welche darauf hinweist, dass die Verifikationsarbeit weiterdelegiert werden kann und dass eine Adresse zur Re-Delegation vorhanden ist. $PKIS_2$ delegiert die Arbeit an $PKIS_1$ weiter. $PKIS_1$ versucht nun einen geeigneten Zertifikatepfad für das Zertifikat des Nutzers aufzubauen. $PKIS_1$ extrahiert dazu die Information über die ausstellende CA aus dem Nutzerzertifikat und sendet eine Anfrage an CA_E und möchte dessen Zertifikate erhalten. CA_E sendet in seiner Antwort zwei Zertifikate zurück, sein öffentliches Schlüsselzertifikat und sein Kreuz-Zertifikat von CA_1. Durch das Kreuz-Zertifikat bemerkt $PKIS_1$, dass einer der Vertrauensanker erreicht ist und somit die Konstruktion des Zertifikatepfades beendet werden kann. Nun verifiziert $PKIS_1$ die Korrektheit des Nutzerzertifikates, des selbst signierten CA_E-Zertifikates und das CA_E-Kreuzzertifikates. Dann fordert der $PKIS_1$ Sperrinformationen über das Nutzerzertifikat von CA_E und über das Kreuz-Zertifikat des CA_E von CA_1 an. Nach Ende des kompletten Verifikationsprozesses sendet $PKIS_1$ das Ergebnis zurück an $PKIS_2$, von wo es an den kontaktierten ISP weitergeht.

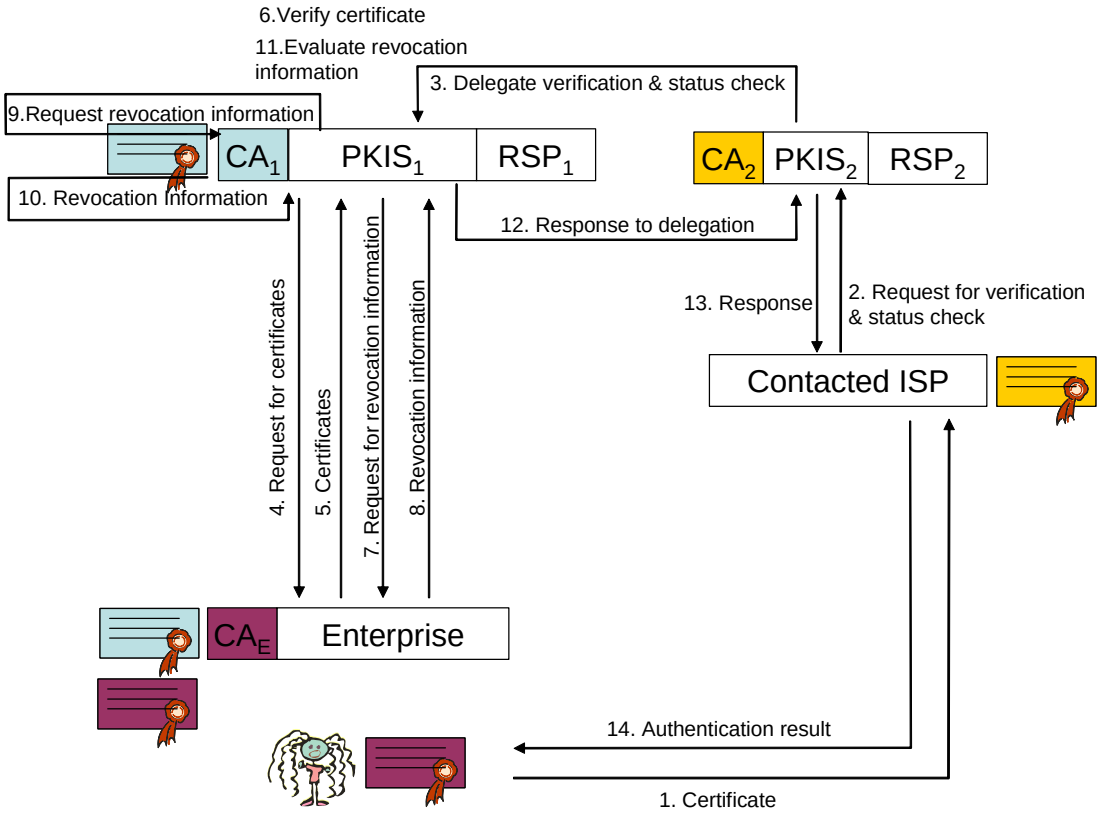


Abbildung 136: Unternehmensnutzerauthentifizierung mit Re-Delegation zu PKISs anderer RSPs

6.4.3 Zertifikateketten im Vergleich

Im Folgenden werden die Zertifikateketten, welche überprüft werden müssen, wenn ein mobiler Nutzer authentifiziert werden soll, dargestellt [EGHH+04]. Das Zertifikat einer Partei X wird als *c(X)* dargestellt. Der Pfeil „$\rightarrow$“ zeigt die Reihenfolge, in der der PKI Server den Pfad konstruiert. Der Pfeil „$\downarrow$“ über einem Zertifikat bedeutet, dass für das entsprechende Zertifikat Rückrufinformationen bzw. Sperrlisten geprüft werden müssen. Diese Sperrinformationen werden von einer Komponente der CA, welche im Pfad an nächster Stelle steht, bereitgestellt. Ein PKI Server muss abgesehen vom ersten Zertifikat, d. h. dem Zertifikat, welches eigentlich überprüft werden soll, und dem letzten Zertifikat in der Kette, das einem Vertrauensanker des PKI Servers zugeordnet wird, alle in den unten dargestellten Ketten dazwischen liegenden Zertifikate herbeiholen.

Nutzer U und kontaktierter ISP gehören zu RSP_i Domäne, U hat c(U) von $CA^{(R)}$ erhalten:

$$\overset{\downarrow}{c}(U) \rightarrow c\left(CA_i^{(R)}\right) \in T\left(PKIS_i^{(R)}\right)$$

Nutzer U und kontaktierter ISP gehören zu RSP_i Domäne, U hat c(U) von $CA^{(I)}$, welche von seinem Heim-ISP betrieben wird, erhalten.

$$\overset{\downarrow}{c}(U) \rightarrow \overset{\downarrow}{c}\left(CA^{(I)}\right) \rightarrow c\left(CA_i^{(R)}\right) \in T\left(PKIS_i^{(R)}\right)$$

Nutzer U gehört zu RSP_i Domäne, kontaktierter ISP gehört zu RSP_j Domäne, U hat c(U) von $CA^{(R)}$ erhalten, RSP Kooperation basiert auf Über-Kreuz-Zertifizierung

$$\overset{\downarrow}{c}(U) \rightarrow \overset{\downarrow}{c}\left(CA_i^{(R)}\right) \rightarrow c\left(CA_j^{(R)}\right) \in T\left(PKIS_j^{(R)}\right)$$

Nutzer U gehört zu RSP_i Domäne, kontaktierter ISP gehört zu RSP_j Domäne, U hat c(U) von $CA^{(I)}$, welche von seinem Heim-ISP betrieben wird, erhalten, RSP Kooperation basiert auf Über-Kreuz-Zertifizierung

$$\overset{\downarrow}{c}(U) \rightarrow \overset{\downarrow}{c}\left(CA^{(I)}\right) \rightarrow \overset{\downarrow}{c}\left(CA_i^{(R)}\right) \rightarrow c\left(CA_j^{(R)}\right) \in T\left(PKIS_j^{(R)}\right)$$

Nutzer U gehört zu RSP_i Domäne, kontaktierter ISP gehört zu RSP_j Domäne, U hat c(U) von $CA^{(R)}$ erhalten, RSP Kooperation basiert auf der Modifikation von T.

$$\overset{\downarrow}{c}(U) \rightarrow c\left(CA_i^{(R)}\right) \in T\left(PKIS_j^{(R)}\right)$$

Nutzer U gehört zu RSP_i Domäne, kontaktierter ISP gehört zu RSP_j Domäne, U hat c(U) von $CA^{(I)}$, welche von seinem Heim-ISP betrieben wird, erhalten, RSP Kooperation basiert auf Modifikation von T

$$\overset{\downarrow}{c}(U) \rightarrow \overset{\downarrow}{c}\left(CA^{(I)}\right) \rightarrow c\left(CA_i^{(R)}\right) \in T\left(PKIS_j^{(R)}\right)$$

Nutzer U gehört zu RSP_i Domäne, kontaktierter ISP gehört zu RSP_j Domäne, U hat c(U) vom RSP erhalten, Kooperation basiert auf Re-Delegation zum $PKIS^{(R)}$

$$\overset{\downarrow}{c}(U) \rightarrow c\left(CA_i^{(R)}\right) \in T\left(PKIS_i^{(R)}\right)$$

Nutzer U gehört zu RSP_i Domäne, kontaktierter ISP gehört zu RSP_j Domäne, U hat c(U) von $CA^{(I)}$, welche von seinem Heim-ISP betrieben wird, erhalten, RSP Kooperation basiert auf Re-Delegation zum $PKIS^{(R)}$

$$\overset{\downarrow}{c}(U) \rightarrow \overset{\downarrow}{c}\left(CA^{(I)}\right) \rightarrow c\left(CA_i^{(R)}\right) \in T\left(PKIS_i^{(R)}\right)$$

Wie man oben deutlich sehen kann, sind die Zertifikatsketten kürzer, wenn die ISPs keine eigenen CAs betreiben. Wenn die Kooperation zwischen den RSPs auf der Modifikation der Menge von Vertrauensankern oder der Re-Delegation beruht, wird der Aufwand im Vergleich zur Über–Kreuz–Zertifizierung reduziert.
Daher ist die Modifikation der Menge der Vertrauensanker die empfehlenswerteste Lösung, wenn man auf einen möglichst schnellen Authentifizierungsvorgang aus ist, da sowohl der Zertifikatepfad minimal ist, als auch weniger Nachrichten im Rahmen des Authentifizierungsvorgangs ausgetauscht werden müssen.

6.5 Zertifikatevalidierung für von Nutzern authentifizierte Parteien

In diesem Abschnitt wird beschrieben, wie mobile Nutzer andere Instanzen, wie z. B. kontaktierte ISPs oder Zugangsnetzbetreiber, authentifizieren. Im Prinzip läuft die Authentifizierung dieser Instanzen auf die gleiche Weise ab, wie in der anderen Richtung. Bei der Verifikation der Zertifikate der Instanzen gibt es aber einen wichtigen Unterschied. Wann immer der Endnutzer den ISP authentifizieren will, kann das Problem auftreten, dass er keinen Internetanschluss hat. Dies bedeutet, dass er keine Sperrinformationen wie CRLs oder andere Statusinformationen erhalten kann und dass er als mobiler Nutzer keine Zertifikatepfade konstruieren kann, die Instanzen beinhalten, die er nicht kennt.

Das Problem der Authentifikation eines kontaktierten ISPs tritt beim Beginn einer Verbindung während des Authentifikationsprozesses auf. Wenn eine Verbindung bereits existiert, z. B. im Falle eines Handovers kann man die alte existierende Verbindung nutzen, um Sperrinformationen über den ISP für die neue Verbindung bereitzustellen. Dazu muss allerdings ein weicher Handover vorliegen. Wenn eine Internetverbindung schon existiert, dann kann der mobile Nutzer Zertifikate und öffentliche Schlüssel der anderen Parteien ähnlich erhalten, wie es für die Nutzerauthentifikation im vorigen Abschnitt beschrieben ist. Da mobile Endgeräte eine begrenzte Kapazität haben, sollte auch hier auf PKIs zurückgegriffen werden. Auf Details der von mobilen Nutzern durchgeführten Authentifikation in dem Fall, dass wenn Internetverbindungen zur Verfügung steht, wird hier jedoch nicht eingegangen. Da die Lösung dieses Problem gegenüber den Lösungsvorschlägen für die Nutzerauthentifikation sehr einfach ist, wird auf das vorherige Kapitel verwiesen. Hier wird jetzt nur das Problem der Authentifikation behandelt, wenn der mobile Nutzer noch keinen Internetanschluss hat. Im Allgemeinen gibt es mehrer Möglichkeiten, dieses Problem anzugehen.

Während die beidseitige Authentifizierung nicht obligatorisch aber wünschenswert ist, gibt es eine Lösung ohne beidseitige Authentifizierung, was heißt, dass der kontaktierte ISP den Endnutzer authentifiziert, der Endnutzer aber den ISP nicht authentifiziert. Der Nachteil dieses Ansatzes ist, dass der Endnutzer keine Garantie über die Identität der Partei hat, mit der er kommuniziert, d. h. Man-in-the-middle Angriffe sind prinzipiell möglich. Dies ist somit keine wirkliche Lösung für die Authentifizierung kontaktierter ISPs.

Ein anderer Weg zur Lösung dieses Problems ist eine begrenzte beidseitige Authentifikation zwischen einem mobilen Nutzer und dem ISP, wobei die Authentifikation des kontaktierten ISP begrenzt ist. Die vom mobilen Nutzer durchgeführte Authentifikation beinhaltet die Anwendung des öffentlichen Schlüssels des kontaktierten ISPs, um die Korrektheit der Authentifikationsinformation zu verifizieren. Der aktuelle Status des Zertifikates des kontaktierten ISP ist dabei jedoch nicht verifiziert. Es kann nicht verifiziert werden, ob das Zertifikat des kontaktierten ISP gesperrt ist. Um

die Korrektheit des Zertifikates des ISP zu verifizieren, ist der öffentliche Schlüssel der ausstellenden CA notwendig. Da keine Sperrinformation über das Zertifikat des ISP ausgewertet wird, ist die Qualität der Authentifzierung begrenzt. Wenn das CA Zertifikat des kontaktierten ISP im Speicher des mobilen Nutzers enthalten ist, kann der mobile Nutzer verifizieren, dass das Zertifikat des ISP von einer zumindest vertrauenswürdigen CA herausgegeben ist. Wenn der öffentliche Schlüssel auf dem Gerät des Nutzers nicht verfügbar ist, dann ist keine Authentifizierung möglich. Dieser Ansatz ist nützlich, wenn Zertifikate, die eine sehr kurze Lebensdauer haben, benutzt werden. Die Authentifizierung ist hierbei jedoch auf Nutzer beschränkt, welche die öffentlichen Schlüssel auf ihren Geräten verfügbar haben. Man kann nicht annehmen, dass die Zertifikate aller kontaktierten ISPs von einer CA ausgestellt werden, die im Speicher des mobilen Nutzers enthalten ist. Wenn sie nicht darin enthalten sind, ist keine Authentifzierung möglich. Damit sind die Sicherheit und die Praktikabilität dieses Ansatzes fragwürdig.

Eine weitere Möglichkeit ist eine verzögerte beidseitige Authentifikation zwischen dem ISP und dem mobilen Nutzer, wenigstens für eine Richtung der Authentifikation. Der mobile Nutzer kann ohne Verzögerung authentifiziert werden. Das heißt, der mobile Nutzer beendet die Authentifikation des kontaktierten ISP zeitlich nach dem Verbindungsaufbau und nicht vorher. Da er einen Internetanschluss für die Konstruktion eines Zertifikatepfades zu einem seiner Vertrauensanker, für die Abfrage der öffentlichen Schlüssel und um Zertifikatestatusinformationen zu erhalten, benötigt, wird die Verifikationsarbeit erst dann gemacht, wenn die Verbindung eingerichtet ist. Dann kann die Verbindung benutzt werden, um die notwendigen Informationen für die Verifikation zu erhalten. Im Falle gesperrter Zertifikate oder von Korrektheitsfehlern bei der Verifikation der Zertifikate kann der mobile Nutzer die Nutzung der Dienste des ISP verweigern und die Verbindung abbrechen. Da das Problem also durch die Verzögerung der Verifikationsarbeit modifiziert ist, bis eine Internetverbindung verfügbar ist, können Lösungen angewandt werden, die ganz ähnlich sind wie die im vorangegangenen Abschnitt. Solche sind jedoch nicht sehr elegant. Sie beinhalten Potential für ISP als Betrüger zu erscheinen und erlauben somit die Kommunikation zwischen Betrügern und einem mobilen Nutzer, bis die Verifikation fertig gestellt ist. Andererseits ist eine solche Lösung billig, da sie keinen zusätzlichen Aufwand für die Zertifikate basierte Authentifikation erfordert, wenn der Nutzer keinen Internetanschluss hat. Im Fall des existierenden Internetanschlusses ist eine Lösung leichter. Eine Lösung für eine Zertifikate basierte Lösung mit existierender Internetverbindung wird in jedem Fall gefordert, beispielsweise im Fall eines Handovers.

Die letzte Möglichkeit ist eine sofortige gegenseitige Authentifikation zwischen Endnutzer und ISP. Da der Endnutzer keine Verbindung hat, mit der er Informationen für die Konstruktion des Zertifikatepfades abfragen und Informationen über die Sperrung des Zertifikates des kontaktierten ISPs erhalten kann, soll bei dieser Lösung der kontaktierte ISP eine Anfrage mit dem Zertifikat des kontaktierten ISPs im Namen des mobilen Nutzers an eine spezielle PKI Komponente senden, welche die gewünschte Arbeit für den mobilen Nutzer ausführt. Diese sendet dann eine Antwort zurück an den kontaktierten

ISP, die dieser an den mobilen Nutzer weitergibt. Die Informationen in dieser Antwort dienen dem mobilen Nutzer dazu, die Authentifikation des kontaktierten ISPs durchzuführen. Die PKI Komponente kann ein PKIS oder einfach ein OCSP Responder sein. Abhängig von dieser Komponente sollte das Anfrage/Antwort-Protokoll entweder dem SCVP Protokoll oder dem OCSP Protokoll folgen. Gegenwärtig gibt es keine Lösung, die es dem mobilen Nutzer erlaubt, das SCVP Protokoll mit einem vertrauensvollen PKIS innerhalb eines Authentifikationsprotokoll zu initiieren. Es ist für den mobilen Nutzer jedoch möglich, OCSP Antworten von einem dezidierten OCSP Responder über den kontaktierten ISP innerhalb des TLS Protokolls, wie es in RFC 3546 beschrieben ist, abzufragen. Die OCSP Antwort muss von einer vertrauensvollen Partei erzeugt werden im Hinblick auf die Erstellung von OCSP Antworten, z.B. von einem OCSP Responder, der vom eigenen RSP betrieben wird. Um den Authentifikationsprozess zu beschleunigen, kann der kontaktierte ISP periodisch OCSP Antworten für sein aktuelles Zertifikat anfordern, z.B. jede Stunde oder jeden Tag, und sie speichern. Dies ist vor allem dann sinnvoll, wenn viele mobile Nutzer demselben OCSP Responder vertrauen. Die Nutzung des OCSP unterstützt den mobilen Nutzer aber nur dabei, Statusinformationen über das Zertifikat des kontaktierten ISP auf effiziente Weise zu erhalten. Das Problem des Nutzers, die Pfadkonstruktion und die Verifikation der Zertifikate, die darin enthalten sind, wird nicht direkt unterstützt. Solange es jedoch keine anderen Lösungen für diese Aufgabe gibt, bleibt nur OCSP übrig. Die OCSP Antworten werden deshalb mit einer modifizierten Semantik strukturiert, die von mobilen Nutzern empfangen und ausgewertet werden können. D.h., für jede OCSP Anfrage führt die kontaktierte PKI Komponente die kompletten Dienste aus, die ein PKIS normalerweise ausführt, statt einer ausschließlichen Überprüfung des Zertifikatestatus werden die Konstruktion des Zertifikatepfades, die Verifikation der Zertifikatekorrektheit und die Überprüfung des Zertifikatestatus durchgeführt. Dann wird die OCSP Antwort vom mobilen Nutzer so interpretiert, als wenn sie eine Antwort eines PKIS wäre. Damit kann der mobile Nutzer die kostspielige Verifizierungsarbeit an einen Server delegieren, ohne selbst eine Internetverbindung zu haben. Auf die Dauer ist jedoch ein Mechanismus erforderlich, der es dem mobilen Nutzer ermöglicht, etwas wie SCVP Anfragen initiieren zu können.

Im Weiteren werden nur Fälle entsprechend der letzten Möglichkeit behandelt. Dabei müssen zwei Fälle unterscheiden werden: Der kontaktierte ISP und der mobile Nutzer gehören zur selben RSP Domäne oder sie gehören zu unterschiedlichen RSP Domänen. Dabei sind wieder normale Teilnehmer und Unternehmensmitarbeiter als Nutzer zu betrachten.

6.5.1 Innerhalb derselben RSP-Domäne

Bei dieser Lösung sendet der Nutzer eine Liste von OCSP Respondern bzw. PKIS an den kontaktierten ISP innerhalb des Authentifikationsprotokolls. Die Liste enthält diejenigen OCSP Responder oder die PKIS, denen der mobile Nutzer hinsichtlich der Korrektheit ihrer Antwort traut. Weiter sollte der mobile Nutzer die öffentlichen Schlüssel der OCSP Responder oder PKIS besitzen, da die OCSP Antwort vom mobilen Nutzer hinsichtlich

ihrer Integrität überprüft werden muss. In dieser Liste sollte der mobile Nutzer auch auf den OCSP Responder oder PKIS verweisen, der von seinem eigenen RSP betrieben wird. Der kontaktierte ISP muss eine OCSP Anfrage nach seinem eigenen Zertifikat generieren und sie an den spezifizierten OCSP Responder bzw. den PKIS schicken.

Im Fall, wenn nur eine reine OCSP Funktionalität angefordert wird, prüft der OCSP Responder ausschließlich den Status des Zertifikates des kontaktierten ISP. Da der kontaktierte ISP sein Zertifikat von der CA erhalten hat, die demselben RSP wie der vertrauensvolle OCSP Responder bzw. PKIS angehört, ist die Statusüberprüfung einfach und schnell. Der OCSP Responder hat nur auf die Quelle der assoziierten CA zuzugreifen. Nach der Statusüberprüfung wird vom OCSP Responder bzw. PKIS eine OCSP Antwort generiert und an den kontaktierten ISP zurückgesandt. Dann kann der kontaktierte ISP die OCSP Antwort an den mobilen Nutzer weiterreichen, der daraufhin deren Integrität verifizieren muss. Da der OCSP Responder bzw. PKIS nur den Status des Zertifikates des kontaktierten ISP überprüft hat, muss der mobile Nutzer die restliche Arbeit für die Zertifikateverifikation selbst machen.

Wenn eine vollständige Zertifikateüberprüfung gewünscht wird, kann die ganze Arbeit an den PKIS delegiert werden. Hierfür erzeugt der kontaktierte ISP eine SCVP Anfrage und sendet diese an den PKIS. Der kontaktierte PKIS extrahiert dann den Herausgeber der Information aus dem Zertifikat. Da angenommen werden kann, dass die demselben RSP zugeordnete CA in der Vertrauenskette des PKIS enthalten ist, ist die Konstruktion des Zertifikatepfades sofort beendet. Damit ist die Korrektheit des Zertifikates verifiziert und der Status des Zertifikates überprüft. In der entsprechenden Antwort schließt der PKIS eine geeignete OCSP Antwort ein und sendet sie an den kontaktierten ISP zurück. Diese OCSP Antwort wird dann an den kontaktierten ISP weitergegeben und von diesem an den mobilen Nutzer, welcher sie nach der Überprüfung ihrer Integrität als vollständiges Verifikationsergebnis interpretiert. Hier muss der mobile Nutzer nur das Ergebnis auswerten. Weitere Verifikationsarbeiten sind nicht erforderlich, da sie vollständig an den PKIS delegiert sind.

Abhängig von der gewählten Variante erhält der mobile Nutzer mit der OCSP Antwort entweder nur ein paar Informationen über den Zertifikatestatus oder eine umfassende Antwort hinsichtlich der Gültigkeit des Zertifikatepfades, kombiniert mit den Statusinformationen.

Da der kontaktierte ISP und der mobile Nutzer derselben RSP Domäne angehören, gibt es keine Schwierigkeiten, einen geeigneten OCSP Responder zu finden. Da der mobile Nutzer den öffentlichen Schlüssel des OCSP Responder bzw. des PKIS seines eigenen RSP kennt, sollte er die OCSP Antwort verifizieren können. Der kontaktierte ISP und ein OCSP-Responder können miteinander über OCSP kommunizieren. Bei einer Kommunikation zwischen kontaktiertem ISP und einem PKIS kann SCVP benutzt werden. Das SCVP Protokoll erlaubt die Kapselung von OCSP Mitteilungen. Die OCSP Anfrage und die Antwort können in den SCVP Erweiterungen platziert werden.

Die Interaktionen beider Varianten werden im Detail jetzt beschrieben.

Wenn nur Statusinformationen über das Zertifikat des kontaktierten ISP gewünscht sind, müssen die Vorgänge eins bis sechs aus Abbildung 137 ausgeführt werden. Die reguläre Fortsetzung im Authentifikationsprotokoll erfolgt dann mit Position sieben. Die Vorgänge sind folgende:

1. Der Nutzer fordert ein von einem vertrauenswürdigen OCSP Responder ausgestelltes OCSP Statement vom kontaktierten ISP an für das Zertifikat des ISP's.
2. Der kontaktierte ISP sendet einen OCSP Request bezüglich seines eigenen Zertifikates zu dem vertrauenswürdigen OCSP Responder. (Dies könnte als Teil eines SCVP Nachrichtenaustausches erfolgen.)
3. Der OCSP Responder enthält von der demselben RSP assoziierten CA bereitgestellte CRL Information.
4. Der OCSP Responder erzeugt ein OCSP Statement und sendet die entsprechende Antwort zum kontaktierten ISP. (Dies kann als Teil eines SCVP Nachrichtenaustauschs erfolgen.)
5. Der kontaktierte ISP sendet die OCSP Antwort zum mobilen Nutzer.
6. Der mobile Nutzer konstruiert den Zertifikatepfad, überprüft die Korrektheit der Zertifikate, und wertet das OCSP statement aus.
7. Der mobile Nutzer fährt mit dem Authentikationsprotokoll fort.

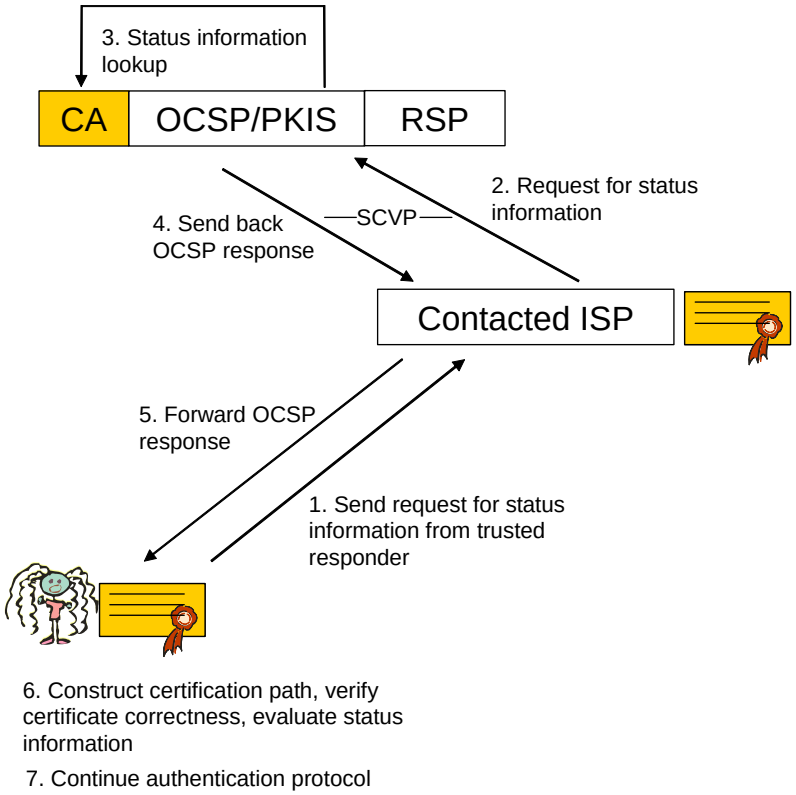


Abbildung 137: Abfrage der Statusinformation des Zertifikats des kontaktierten ISP

Außer der Statusinformation des Zertifikates des kontaktierten ISP können weitere Arbeiten wie die Konstruktion des Zertifikatepfades und die Verifikation der Korrektheit für den mobilen Nutzer durchgeführt werden, d.h. die gesamte Verifikationsarbeit wird an eine vertrauensvollen PKIS delegiert. Um dies durchführen zu können, wird die Semantik eines OCSP Anweisung neu definiert. Der OCSP Status „good“ bedeutet nun, dass das Zertifikat nicht nur nicht gesperrt ist, sondern dass es auch korrekt ist. Die Interaktionen sind in Abbildung 138 dargestellt. Der Ablauf erfolgt mit den Positionen eins bis sechs; die reguläre Fortsetzung mit Position sieben:

1. Der Nutzer stellt eine Anfrage an den kontaktierten ISP dahingehend, dass dieser die Delegation des kompletten Handshakes zu einem vertrauenswürdigen PKIS weiterleitet.
2. Der kontaktierte ISP sendet eine Verifikationsanfrage an bezüglich seines eigenen Zertifikates zum vertrauenswürdigen PKIS. (Dies könnte als Teil eines SCVP oder OCSP Nachrichtenaustausches erfolgen.)
3. Der PKIS konstruiert den Zertifikatepfad für das Zertifikat des kontaktierten ISPs und verifiziert die Korrektheit der Zertifikate im Pfad. Zusätzlich überprüft der PKIS den Status der Zertifikate.
4. The PKIS erzeugt ein OCSP Statement —unter Berücksichtigung der veränderten Semantik— und sendet die entsprechende Antwort zum kontaktierten ISP. (Dies kann als Teil eines SCVP Nachrichtenaustauschs erfolgen.)
5. Der kontaktierte ISP sendet eine OCSP Antwort zum mobilen Nutzer.
6. Der mobile Nutzer wertet das OCSP Statement aus.
7. Der mobile Nutzer fährt mit dem Authentifizierungsprotokoll fort.

Diese Variante ist vorzuziehen, wenn die Kapazität des mobilen Endgerätes begrenzt ist. Sie ist weiter vorzuziehen, wenn die Konstruktion des Zertifikatepfades kompliziert ist, d.h. wenn diese Konstruktion einen Internetanschluss für den Nutzer erfordert, um mehrere CA Zertifikate von unterschiedlichen CAs aus dem Zertifikatepfad zu erhalten. Die Zertifikatestruktur wurde jedoch so konstruiert, dass keine komplexen Zertifikatepfade für ISP-Zertifikate vorliegen, wenn der mobile Nutzer und der kontaktierte ISP derselben ISP Domäne angehören.

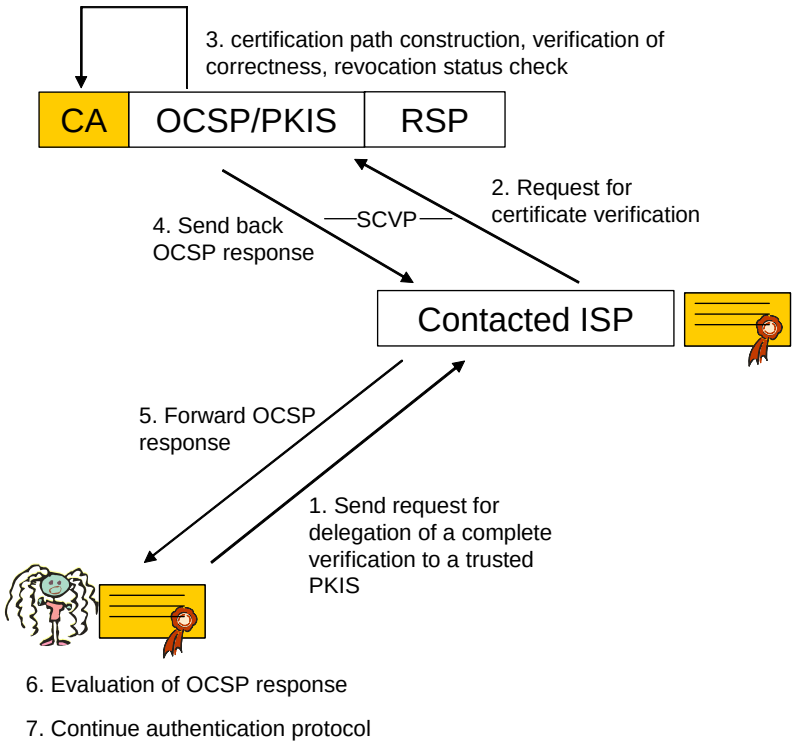


Abbildung 138: Delegation der Verifikation des Zertifikates des kontaktierten ISP

6.5.2 Unterschiedliche RSP Domänen

Die Tatsache, dass der mobile Nutzer und der kontaktierte ISP unterschiedlichen RSP Domänen angehören, hat beträchtliche Folgen. Eine der Wesentlichen ist es, dass es nicht sinnvoll ist, das reine OCSP Protokoll anzuwenden, wenn der mobile Nutzer ausschließlich mit der Statusinformation bezüglich des Zertifikates des kontaktierten ISP zu versorgen ist. Stattdessen wird mit einem semantisch modifizierten OCSP Protokoll gearbeitet, d.h. das OCSP Protokoll wird dazu benutzt, den mobilen Nutzer mit den Ergebnissen der vollständigen Zertifikateverifikation, die vom vertrauenswürdigen PKIS erzeugt wurden, zu versorgen.

Bevor die Lösung für den Fall der unterschiedlichen RSP Domänen im Detail beschrieben wird, wird kurz begründet, warum es für mobile Nutzer nicht ausreichend ist, mit der puren Statusinformation über das Zertifikat des kontaktierten ISP versorgt zu werden. Es gibt hierfür mehrere Gründe. Es sei der RSP des mobilen Nutzers im weiteren RSP_1 und der des kontaktierten ISP RSP_2. Damit erhält der kontaktierte ISP sein Zertifikat von einer CA_2, die von RSP_2 betrieben wird. Es muss dabei angenommen werden, dass der mobile Nutzer weder RSP_2 noch CA_2 kennt und diese somit keinen Vertrauensanker für ihn darstellt. Wenn der mobile Nutzer nun versucht, die Verifikationsarbeit selbst ohne Hilfe eines PKIS durchzuführen, muss als erstes ein Zertifikatepfad vom Zertifikat des kontaktierten ISP zu einem seiner Vertrauensanker hergestellt werden. Dies erfordert, dass der mobile Nutzer, nachdem er mit dem Zertifikat des kontaktierten ISPs versorgt ist,

die Zertifikate der CA_2 erhält. Die erfolgreiche Konstruktion eines Zertifikatepfades ist für den mobilen Nutzer nur garantiert, wenn die CA von RSP_1 mit der CA von RSP_2 kreuz-zertifiziert ist unter der zusätzlichen Annahme, dass CA_1 zur Vertrauensankerkette des mobilen Nutzers gehört. Ohne diese Kreuz-Zertifizierung kann nicht angenommen werden, dass der mobile Nutzer einen brauchbaren Zertifikatepfad konstruieren kann. Für die Konstruktion des Zertifikatepfades benötigt der mobile Nutzer jedoch die Zertifikate der CA_2. Diese kann er jedoch nur erhalten, wenn er eine Internetverbindung hat. Weiter ist die Sperrinformation über das Zertifikat des kontaktierten ISP allein nicht hinreichend. Um den vollständigen Zertifikatepfad für das Zertifikat des kontaktierten ISP validieren zu können, ist auch die Sperrinformation über die Kreuz-Zertifizierung erforderlich. Aber auch dies erfordert einen Internetanschluss für den mobilen Nutzer. Hieraus folgt, dass die Arbeit der vollständigen Zertifikateverifikation von einem mobilen Nutzer nicht selbst ausgeführt werden kann, wenn er keinen Internetanschluss hat, und somit sollte diese Arbeit auch unter dem Aspekt der begrenze Kapazität des Endgerätes delegiert werden.

Die im Folgenden dargestellten Lösungen beruhen auf den gleichen Modifikationen der Semantik der OCSP Mitteilungen, wie dies im vorigen Abschnitt vorgeschlagen wurde. Auch die grundsätzlichen Prinzipien für dien Interaktion der beteiligten Parteien sind die gleichen wie im vorigen Abschnitt. Zwei Lösungen werden vorgeschlagen, die sich im Nachrichtenfluss unterscheiden. In der ersten Lösung delegiert der kontaktierte RSP die Arbeit an den $PKIS_1$, der vom RSP_1 versorgt wird. In der zweiten Lösung wird die Arbeit an den $PKIS_2$ delegiert, der vom RSP_2 versorgt wird.

In der ersten Lösung wird die gesamte Interaktion vom mobilen Nutzer dadurch ausgelöst, dass er eine Anfrage an den kontaktierten ISP sendet. Mit dieser Anfrage wird der kontaktierte ISP informiert, dass der mobile Nutzer eine Verifikation des Zertifikates des kontaktierten ISP durch eine Vertrauensinstanz anfordert, welche $PKIS_1$ ist. Dann sendet der kontaktierte ISP eine OCSP Anfrage nach seinem eigenen Zertifikat an den $PKIS_1$. Diese Anfrage kann innerhalb eines SCVP gekapselt sein. Nun führt der $PKIS_1$ die komplette Zertifikateverifikationsarbeit für den mobilen Nutzer aus. D. h. der $PKIS_1$ konstruiert den Zertifikatepfad für das Zertifikat des kontaktierten ISP. Dieser Pfad kann abhängig von der Definition der Menge der Vertrauensanker des $PKIS_1$ variieren. Er kann entweder aus dem Zertifikat des gerade kontaktierten ISP bestehen, wenn CA_2 in der Menge der Vertrauensanker von $PKIS_1$ enthalten ist, oder aus dem Zertifikat des kontaktierten ISP und den Zertifikaten vom CA_2, wenn CA_2 nicht in der Vertrauensankerkette von $PKIS_1$ nicht enthalten ist. Dann verifiziert $PKIS_1$ die Korrektheit dieser Zertifikate und überprüft als nächstes deren Status. Für die Überprüfung des Status des Zertifikates des kontaktierten ISP fordert $PKIS_1$ Sperrinformationen von der CA_2 an. Hierfür gibt es einige Möglichkeiten. Wenn man annimmt, dass alle oder einige der RSP ihre Sperrlisten in regelmäßigen Zeitintervallen austauschen, hat $PKIS_1$ die entsprechenden Sperrlisten eventuell lokal verfügbar. Wenn der $PKIS_1$ die geforderte Sperrliste nicht gespeichert hat, hat er die spezielle Sperrinformation von der CA_2 anzufordern. Dies kann wieder über einen Austausch von OCSP Nachrichten zwischen dem $PKIS_1$ und der CA_2 durchgeführt werden. Es wird angenommen, dass RSP und ihre Komponenten so zusammenarbeiten, dass ein solcher

Mitteilungsaustausch unterstützt wird. Wenn die CA_2 eine geeignete OCSP Antwort zurückgegeben hat, kann der $PKIS_1$ eine neue OCSP Antwort mit modifizierter Semantik erzeugen und sie an den kontaktierten ISP senden. Der kontaktierte ISP sendet dann die Antwort an den mobilen Nutzer. Dieser evaluiert die Antwort und fährt mit den Authentifikationsprotokoll fort. Die vollständigen Interaktionen der involvierten Parteien sind in Abbildung 139 zusammengefasst.

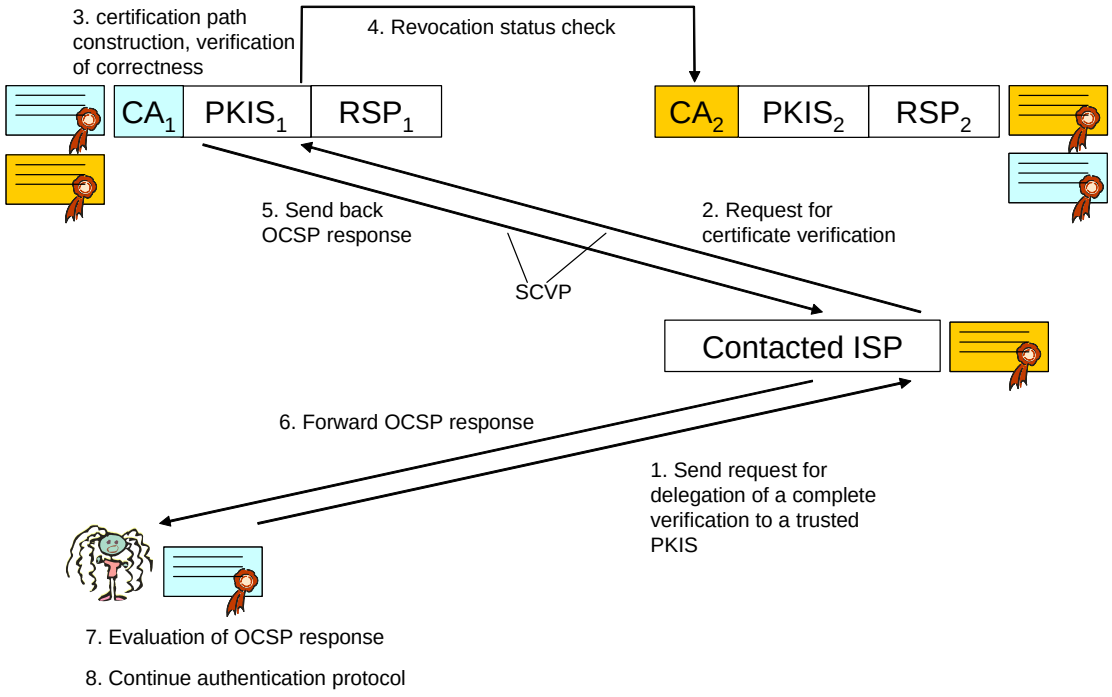


Abbildung 139: Delegation der Verifikation innerhalb unterschiedlicher RSP Domänen

Die zweite Lösung unterscheidet sich von der ersten darin, dass der kontaktierte ISP mit dem $PKIS_1$ nicht kommuniziert. Stattdessen kontaktiert er den $PKIS_2$, der in seiner eigenen RSP Domäne bereitgestellt wird. Der $PKIS_2$ kontaktiert dann den $PKIS_1$, um nach der angeforderten OCSP Antwort mit der modifizierten Semantik nachzufragen. Wenn der $PKIS_1$ seine Verifikationsarbeit insgesamt ausgeführt hat, sendet er die gewünschte OCSP Antwort an den $PKIS_2$ ab. Dieser gibt die Antwort an den kontaktierten ISP weiter, der sie zum mobilen Nutzer leitet. Der Nachrichtenfluss dieser Lösung ist in Abbildung 140 gezeigt.

Vergleicht man beide Lösungen, stellt man bei der zweiten Lösung einen zusätzlichen Umweg fest, und somit eine geringere Effizienz. In einigen Fällen kann es jedoch erforderlich sein, eine solche Lösung anzuwenden, z. B., wenn PKIS ihre Dienste nur anderen RSP oder Instanzen, die zur eigenen RSP Domäne gehören, anbieten.

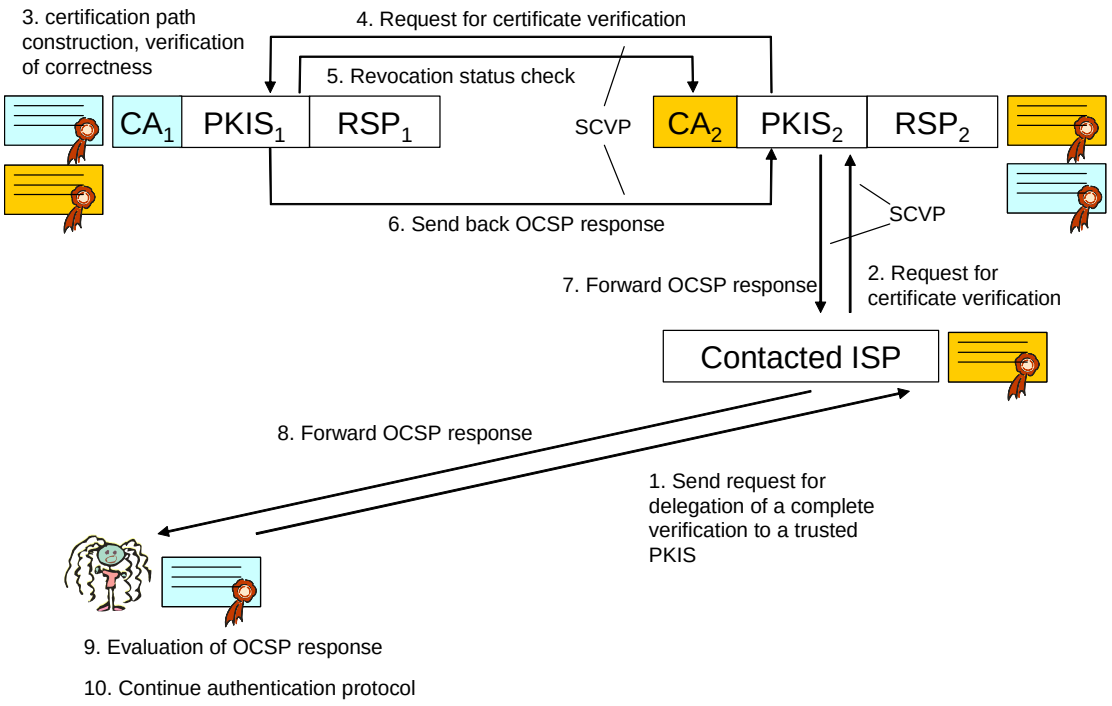


Abbildung 140: Umgeleitete Delegation der Verifikation bei unterschiedlichen RSP Domänen

6.5.3 Zertifikateketten im Vergleich

Im Folgenden werden die Zertifikateketten, welche überprüft werden müssen, wenn ein mobiler Nutzer eine Partei authentifiziert, dargestellt [EGHH+04]. Die Notation entspricht der aus Abschnitt 6.4.3. Es können sich je nach Zugehörigkeit zur RSP-Domäne und Art der Kooperation der RSPs untereinander die Folgenden Zertifikatsketten ergeben:

Nutzer U und der kontaktierte ISP gehören zur Domäne des RSP_i. hat c(ISP) von $CA_i^{(R)}$ erhalten.

$$\overset{\downarrow}{c}(ISP) \rightarrow c\left(CA_i^{(R)}\right) \in T\left(PKIS_i^{(R)}\right)$$

Nutzer U und der kontaktierte ISP gehören zur Domäne des RSP_i. ISP hat c(ISP) von $CA^{(I)}$ erhalten, welche von ihm selbst betrieben wird.

$$\overset{\downarrow}{c}(ISP) \rightarrow \overset{\downarrow}{c}\left(CA^{(I)}\right) \rightarrow c\left(CA_i^{(R)}\right) \in T\left(PKIS_i^{(R)}\right)$$

Nutzer U gehört zu RSP_i Domäne und der kontaktierte ISP gehört zu RSP_j Domäne, ISP hat c(ISP) von $CA_j^{(R)}$ erhalten. Die RSP Kooperation basiert auf Über-Kreuz-Zertifizierung.

$$\overset{\downarrow}{c}(ISP) \rightarrow \overset{\downarrow}{c}\left(CA_j^{(R)}\right) \rightarrow c\left(CA_i^{(R)}\right) \in T\left(PKIS_i^{(R)}\right)$$

Nutzer U gehört zu RSP_i Domäne und der kontaktierte ISP gehört zu RSP_j Domäne, ISP hat c(ISP) von $CA^{(I)}$ erhalten, welche von ihm selbst betrieben wird. Die RSP Kooperation basiert auf Über-Kreuz-Zertifizierung.

$$\overset{\downarrow}{c}(ISP) \rightarrow \overset{\downarrow}{c}\left(CA^{(I)}\right) \rightarrow \overset{\downarrow}{c}\left(CA_j^{(R)}\right) \rightarrow c\left(CA_i^{(R)}\right) \in T\left(PKIS_i^{(R)}\right)$$

Nutzer U gehört zu RSP_i Domäne und der kontaktierte ISP gehört zu RSP_j Domäne, ISP hat c(ISP) von $CA_j^{(R)}$ erhalten. Die RSP Kooperation basiert auf der Modifikation von *T*.

$$\overset{\downarrow}{c}(ISP) \rightarrow c\left(CA_j^{(R)}\right) \in T\left(PKIS_i^{(R)}\right)$$

Nutzer U gehört zu RSP_i Domäne und der kontaktierte ISP gehört zu RSP_j Domäne, ISP hat c(ISP) von $CA^{(I)}$ erhalten, welche von ihm selbst betrieben wird. Die RSP Kooperation basiert auf der Modifikation von *T*.

$$\overset{\downarrow}{c}(ISP) \rightarrow \overset{\downarrow}{c}\left(CA^{(I)}\right) \rightarrow c\left(CA_j^{(R)}\right) \in T\left(PKIS_i^{(R)}\right)$$

Nutzer U gehört zu RSP_i Domäne und der kontaktierte ISP gehört zu RSP_j Domäne, ISP hat c(ISP) von $CA_j^{(R)}$ erhalten. Die RSP Kooperation basiert auf Re-Delegation zum $PKIS_j^{(R)}$

$$\overset{\downarrow}{c}(ISP) \rightarrow c\left(CA_j^{(R)}\right) \in T\left(PKIS_j^{(R)}\right)$$

Nutzer U gehört zu RSP_i Domäne und der kontaktierte ISP gehört zu RSP_j Domäne, ISP hat c(ISP) von $CA^{(I)}$ erhalten, welche von ihm selbst betrieben wird. Die RSP Kooperation basiert auf Re-Delegation zum $PKIS_j^{(R)}$

$$\overset{\downarrow}{c}(U) \rightarrow \overset{\downarrow}{c}\left(CA^{(I)}\right) \rightarrow c\left(CA_j^{(R)}\right) \in T\left(PKIS_j^{(R)}\right)$$

6.6 Konklusion

In diesem Kapitel wurde eine PKI-basierte Authentifizierungslösung entwickelt. Wenn man Zertifikate für die Authentifizierung einsetzt, muss man ganze Zertifikateketten verifizieren. In der Welt der Mobilkommunikation existieren spezielle Randbedingungen. Mobile Endgeräte haben z.B. oft eine vergleichsweise begrenzte Rechenkapazität. Hier wird eine PKI-basierte Lösung vorgeschlagen. PKI-Server ermöglichen die Delegation der Überprüfung der Zertifikateketten von Endgeräten an die PKI-Server, um diesen Vorgang zu beschleunigen. Des Weiteren wird eine spezielle Struktur der PKI-Komponenten vorgeschlagen, welche es ermöglicht kurze Zertifikatspfade zu erzwingen. Des Weiteren wird eine Lösung für das Problem eines Nutzers aufgezeigt, der keinen Internet-Zugang während der Authentifizierungsphase hat. Diese ist u. a. später in Kapitel 9 auch implementiert.

Nachdem hier die PKI basierte Lösung für die Authentifizierung vorgenommen wurde, wird im nächsten Kapitel eine für das hier zu Grunde gelegte Szenario eines umherwandernden „roamenden“ Nutzers, der z.B. auf Ressourcen des Unternehmens, für das er arbeitet, zugreifen will, eine zu den Modellen aus Kapitel 3 passende Autorisierungslösung entwickelt.

7 Autorisierung

Wenn ein Nutzer Zugriff auf Ressourcen gleich welcher Art haben möchte, entscheidet die Partei, welche diese Ressourcen verwaltet, darüber, ob dem Nutzer Zugriff gewährt wird oder nicht. Diese Entscheidung fußt auf der Strategie der die Ressourcen verwaltenden Partei. Eine solche Strategie besteht aus einer Menge von Autorisierungsregeln. Diese Strategie kann für verschiedene Nutzer unterschiedliche Regeln beinhalten. Derartige Regeln sind üblicherweise im Detail oder implizit in einem Vertrag festgelegt. Dies gilt auch für ISPs, die ihren Kunden Ressourcen für den Zugang zum Internet bereitstellen. Wenn ein Nutzer von einem ISP Zugang zum Internet verlanget, dann entscheidet der ISP, ob der Nutzer autorisiert ist Zugang zum Internet zu erhalten oder nicht.

Die Situation wird schwieriger, wenn es kein vertragliches Verhältnis zwischen der Partei, die die Ressourcen zu Verfügung stellt und der die sie nutzen möchte, gibt, so wie es bei einem kontaktierten ISP und bei einem umherwandernden Nutzer, der auf seinen Reisen von einem beliebigen ihm unbekannten ISP im Rahmen eines Roaming Dienstes Zugang zum Internet haben möchte. Die Autorisierungsregeln des Heim-ISP und des kontaktierten ISP des Nutzers können sich unterscheiden. Wünschenswert wäre jetzt, dass der Nutzer auf dieselbe Weise auf die Ressourcen des kontaktierten ISP zugreifen kann, wie er auf die Ressourcen seines Heim-ISP zugreifen kann, sofern dies nicht im Widerspruch zur Strategie des kontaktierten ISP steht. Da ein Nutzer versuchen kann Dienste bzw. Ressourcen zu nutzen, für die er gemäß des mit seinem Heim-ISP abgeschlossenen Vertrages keine Berechtigung hat, muss ein Mechanismus geschaffen werden, der es einem kontaktierten ISP ermöglicht die Nutzung der Ressourcen auf Basis der Autorisierungsregeln des Heim-ISP eines Nutzers durchzusetzen. Dies bedeutet einen Austausch der Autorisierungsregeln zwischen den beiden ISPs.

Die Situation wird noch schwieriger, wenn der Heim-ISP und der kontaktierte ISP keinerlei vertragliche Beziehungen zueinander haben. Womöglich sind sie einander unbekannt. Dafür ist die Rolle des Roaming Service Providers (RSPs) entscheidend. Der RSP ist im Rahmen des Roaming dafür verantwortlich, die kontaktierten ISPs mit den für die Autorisierung relevanten Daten zu versorgen. Der RSP kann diese Daten selbst verteilen oder eine ihm assoziierte Partei wie z.B. eine CA kann dies tun. Auf Basis der Autorisierungsinformation, welche vom RSP oder der ihm assoziierten Partei bereitgestellt wird, setzt der kontaktierte ISP dann die Autorisierung durch.

Es gibt unterschiedliche Möglichkeiten das Autorisierungsproblem im Rahmen des Roaming zu lösen. Die Anforderungen der Autorisierungslösung für das Roaming beinhalten zum Teil Anforderungen für allgemeine AAA Modelle, wie sie in [FVCG00] dargestellt sind. Die hier entwickelte Lösung muss darüber hinaus den Anforderungen der in Kapitel 3 beschriebenen Geschäftsmodelle gerecht werden.

Die hier entwickelte Lösung verwendet die Security Assertions Markup Language (SAML) so wie sie in [SAML02a, SAML02b, SAML02c, SAML02d] beschrieben ist. Sie wird für den Austausch von für die Autorisierung relevanten Informationen zwischen dem RSP und dem kontaktierten ISP verwendet. Prinzipiell könnte mit der Extended Markup Language (XML) auch eine proprietäre Lösung modelliert werden, da jedoch für die in dieser Arbeit entwickelte Lösung, wie Eingangs bereits gesagt, möglichst existierende Standards eingesetzt werden sollen, wird hier auf SAML als existierenden Standard zurückgegriffen. Während SAML keine Mechanismen zur Zugangskontrolle, wie z. B. nach IEEE 802.1x beinhaltet und daher für die Authentifizierung im Rahmen der Zugangskontrolle nicht geeignet ist, ist es zur Weiterleitung von Authentifizierungs und Autorisierungsinformation ideal. Um Autorisierungsentscheidungen zu treffen, können Standardtechnologien, wie Access Control Lists (ACLs) oder Role Based Access Control (RBAC) [FSGK01] eingesetzt werden.

Im folgenden Abschnitt 7.1 werden die Anforderungen, welche die Autorisierungslösung erfüllen soll, aufgestellt. Danach wird in Abschnitt 7.2 die Architektur der auf SAML basierenden hier entwickelten Lösung dargestellt. In Abschnitt 7.3 wird noch eine weitere Lösung, die auf eine besonders schnelle Autorisierung abzielt, dargestellt.

7.1 Anforderungen an die Autorisierungsarchitektur

Die Anforderungen an die Autorisierung im Rahmen der Roamingarchitektur sind folgende:

- Effizienz der Autorisierungsentscheidung und Durchsetzung: Die benötigte Zeit für den gesamten Autorisierungsprozess sollte gering sein. Das bedeutet, dass die Zeit für den Aufbau der Verbindungen gering sein sollte, und das Datenvolumen, welches übertragen werden muss, sollte ebenfalls möglichst gering sein. Die Zeit für die Entscheidung selbst und die aus der Entscheidung resultierenden durchzuführenden Maßnahmen d.h. die Durchsetzung sollte ebenfalls möglichst gering sein.
- Aktualität der Autorisierungsregeln: Der Heim-ISP wird von Zeit zu Zeit vielleicht die seine Kunden betreffenden Regeln ändern wollen. Dementsprechend ändert sich, was für den Nutzer erlaubt oder verboten ist. Derartige Änderungen sollten bei Zugriffskontrollentscheidungen möglichst schnell berücksichtigt werden. Dies betrifft sowohl die Sicherheit als auch die Funktionalität.
- Unterstützung von Flexibilität bezüglich der Zugriffskontrollregeln: Wenn die Autorisierungsregeln geändert werden, müssen die Kosten für die Durchsetzung der Veränderungen so niedrig wie möglich sein.
- Privatsphäre und Geheimhaltung: Der Nutzer hat das Interesse, dass bei dem Autorisierungsprozess der kontaktierte ISP so wenig Information wie möglich bezüglich seiner Person erhält. Der Heim ISP kann ebenfalls das Interesse haben, dass der kontaktierte ISP so wenig wie möglich an Information über den Nutzer, über ihn und über das zugrunde liegende Vertragsverhältnis erhält, da der kontaktierte ISP sein Konkurrent sein könnte.

- Protokoll: Das Protokoll, welches für die Kommunikation verwendet wird, muss flexibel genug sein, alle möglichen Autorisierungsinformationen, welche benötigt werden, um eine Autorisierungsentscheidung zu treffen, berücksichtigen zu können. Es muss dem Rechnung tragen, dass die Parteien, welche die Autorisierungsentscheidungen treffen und die, welche sie durchsetzen, zu unterschiedlichen Domänen gehören. Das Protokoll muss den Transport von Authentifikations- und Autorisierungsinformation in Kombination erlauben. Das Protokoll sollte in der Lage sein mit "Zwischenhändlern" zu Recht zukommen, wenn bei der Autorisierung eine Kette von Einheiten involviert ist.
- Sicherheit bei der Übertragung der Autorisierungsinformation: Es muss gewährleistet sein, dass die Autorisierungsinformation unverfälscht und geschützt vor dem Zugriff Dritter übertragen wird.
- Verfall und Widerruf von Zugriffsrechten: In manchen Fällen kann es nötig sein, dass Autorisierungsentscheidungen oder Regeln auf welchen Autorisierungsentscheidungen basieren, geändert werden. Es sollte daher einen Mechanismus geben, die Entscheidungen zu widerrufen. Dies ist besonders dann wichtig, wenn Abrechnung mit der Autorisierung verbunden wird.

Diese Anforderungen werden in der hier beschriebenen Lösung erfüllt.

Bevor die Architektur beschrieben wird, zunächst noch ein Exkurs, der verdeutlicht, dass die Architektur alleine keine definierte maximale Zeit für einen Vorgang sicherstellen kann.

7.1.1 Geschwindigkeit von Protokollabläufen

Einen genauen Wert für die Geschwindigkeit, welche der Authentifizierungs- oder der Autorisierungsvorgang benötigen, anzugeben ist schwierig, da dies von verschiedenen Faktoren abhängt:

- Leistungsfähigkeit des Endgerätes
- Leistungsfähigkeit des Servers
- Leistungsfähigkeit des Netzes

Wenn man die für das Erreichen einzelner Punkte benötigte Zeit in Form von Antwortzeiten, die beim kontaktieren verschiedener Server entstehen, miteinander vergleicht, erkennt man, dass es grosse Unterschiede hinsichtlich der Zeiten gibt. Im Folgenden wird von demselben Rechner aus einmal Yahoo in China, Deutschland und in den U.S.A. kontaktiert. Dies ergibt folgendes Bild:

```
C:\>ping cn.yahoo.com

Ping cn.vip.cnb.yahoo.com [202.43.216.42] mit 32 Bytes Daten:

Antwort von 202.43.216.42: Bytes=32 Zeit=505ms TTL=231
Antwort von 202.43.216.42: Bytes=32 Zeit=505ms TTL=231
Antwort von 202.43.216.42: Bytes=32 Zeit=503ms TTL=231
Antwort von 202.43.216.42: Bytes=32 Zeit=507ms TTL=231
```

```
Ping-Statistik für 202.43.216.42:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0 (0% Verlust),
Ca. Zeitangaben in Millisek.:
    Minimum = 503ms, Maximum =  507ms, Mittelwert =  505ms

C:\>ping www.yahoo.de

Ping www.euro.yahoo.akadns.net [217.12.3.11] mit 32 Bytes Daten:

Antwort von 217.12.3.11: Bytes=32 Zeit=24ms TTL=237
Antwort von 217.12.3.11: Bytes=32 Zeit=25ms TTL=238
Antwort von 217.12.3.11: Bytes=32 Zeit=25ms TTL=237
Antwort von 217.12.3.11: Bytes=32 Zeit=24ms TTL=238

Ping-Statistik für 217.12.3.11:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0 (0% Verlust),
Ca. Zeitangaben in Millisek.:
    Minimum = 24ms, Maximum =  25ms, Mittelwert =  24ms

C:\>ping www.yahoo.com

Ping www.yahoo.akadns.net [216.109.117.110] mit 32 Bytes Daten:

Antwort von 216.109.117.110: Bytes=32 Zeit=93ms TTL=50
Antwort von 216.109.117.110: Bytes=32 Zeit=93ms TTL=50
Antwort von 216.109.117.110: Bytes=32 Zeit=94ms TTL=49
Antwort von 216.109.117.110: Bytes=32 Zeit=93ms TTL=49

Ping-Statistik für 216.109.117.110:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0 (0% Verlust),
Ca. Zeitangaben in Millisek.:
    Minimum = 93ms, Maximum =  94ms, Mittelwert =  93ms
```

Man erkennt deutlich, dass Yahoo in Deutschland sehr viel schneller von dem in Deutschland befindlichen Rechner aus erreicht wird, als Yahoo in den USA und China. Wenn man gezielt die Mittelwerte gegenüberstellt, sieht man, dass die mittlere Antwortzeit für China mit 505ms fast 21 mal so groß ist, wie die für Deutschland mit 24ms und die mittlere Antwortzeit nach U.S.A. ist mit 94 ms fast 4 mal so groß. Dies liegt u. a. im Wesentlichen an dem deutlich längeren Weg, den die Pakete zurücklegen müssen. Auch das Übertragungsmedium spielt hierbei eine Rolle. Je nach Übertragungsmedium ergeben sich folgende Verzögerungszeiten:

Tabelle 5: Verzögerung nach Übertragungsmedium

Medium	**Verzögerung**
Coax – Kabel	0.004 ms pro km
Optisches Fiberglas	0.005 ms pro km
Coax - Kabel (Unterwasser)	0.006 ms pro km
Satellit in ca. 14.000km Höhe	110 ms
Satellit in ca. 36.000km Höhe	260 ms

Idealerweise sollten Verzögerungen so gering sein, dass sie Echtzeitanwendungen möglich machen. Bei VoIP gelten nach [ITUT03] Verzögerungen von über 300 ms als nicht mehr tolerierbar, da sie einen Dialog deutlich behindern. Abgesehen von der Entfernung und dem Übertragungsmedium ist die Leistungsfähigkeit des Endgerätes und der verwendeten Software, was bei VoIP beispielsweise die eingesetzten Codecs bedeutet, von entscheidender Bedeutung, da die verwendete Software i. d. R. unterschiedliche Eigenschaften, wie z.B. die benötigte Rechenleistung aufweist.

Neben den eingesetzten Algorithmen hat auch die grundlegende hardwarebedingte Leistungsfähigkeit der beteiligten Komponenten entscheidenden Einfluss u. a. auf die Geschwindigkeit mit der Authentifizierungs- und Autorisierungsvorgänge ablaufen. Die folgende Tabelle zeigt beispielhaft Ausführungszeiten der Algorithmen RSA und ECC auf den beiden Mikrocontrollern ATmega128 und CC1010, die in vergleichsweise kleinen sparsamen Geräten zum Einsatz kommen können:

Tabelle 6: Durchschnittliche ECC und RSA Ausführungszeit[Gura04]

Algorithm	**ATmega128 @ 8MHz**			**CC1010 @ 14.7456MHz**		
	time	data mem	code	time	data mem	code
	s	bytes	bytes	s	ext+int, bytes	bytes
ECC secp160r1	0.81s	282	3682	4.58s	180+86	2166
ECC secp192r1	1.24s	336	3979	7.56s	216+102	2152
ECC secp224r1	2.19s	422	4812	11.98s	259+114	2214
Mod. exp. 512	5.37s	328	1071	53.33s	321+71	764
RSA-1024 public-key $e = 2^{16} + 1$	0.43s	542	1073	> 4.48s		
RSA-1024 private-key w. CRT	10.99s	930	6292	~ 106.66s		
RSA-2048 public-key $e = 2^{16} + 1$	1.94s	1332	2854			
RSA-2048 private-key w. CRT	83.26s	1853	7736			

Man erkennt deutlich, dass sowohl der eingesetzte Algorithmus als auch der verwendete Mikrocontroller die Ausführungszeit stark beeinflussen. Wenn man den ATmega128 betrachtet erreicht man Zeiten zwischen 0,43s und 83,26s je nach Algorithmus. Wenn man z. B. den Algorithmus ECCsecp224r1 betrachtet, dann variieren die Zeiten je nach Mikrocontroller von 2,19s zu 11,98s.

Wenn man ein High-End Laptop der neuesten Generation neben ein altes Handy stellt, werden die Unterschiede noch größer ausfallen. Damit ist klar, dass es große Unterschiede zwischen den beteiligten Endgeräten geben kann, was die Leistungsfähigkeit angeht. Auf Grund dessen und der oben beschriebenen Einflussfaktoren ist es nicht möglich eine konkrete Zeit vorzugeben. Es wäre ideal eine maximale Verzögerung von einer halben Sekunde zu erreichen. Dies kann jedoch mit der in dieser Arbeit entwickelten Architektur nicht ungeachtet der eingesetzten Hardware garantiert werden. Wenn man beispielsweise von einem aktuellen handelsüblichen WLAN AP, Radius Server und als mobilem Endgerät einem aktuellen Laptop ausgeht,

sollte innerhalb derselben Domäne die Grenze von 0,3 Sekunden nicht überschritten werden – weder bei einem Authentifizierungs- noch bei einem Autorisierungsvorgang.

7.2 Autorisierungsarchitektur

Im Folgenden werden die Architektur und das grundlegende Prinzip der Autorisierungslösung beschrieben. Aus den unterschiedlichen in [LGGV00] dargestellten Architekturen wird das „pull“ Modell für die Autorisierungslösung ausgewählt und an die hier relevanten Rollen aus Kapitel 3.1 angepasst. Die daraus entstehende Architektur und die Funktion der einzelnen Komponenten ist in der folgenden Abbildung 141 dargestellt.

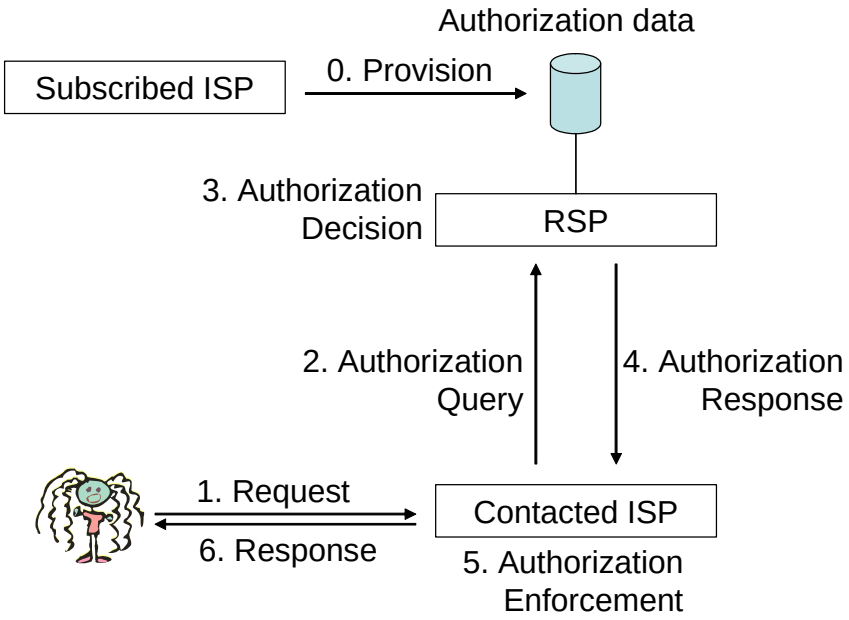


Abbildung 141: Autorisierungsprozess im „pull“-Modell

Unter einer geschlossenen Strategie wird in dieser Arbeit verstanden, dass der Gebrauch von Ressourcen, welcher nicht ausdrücklich erlaubt ist, verboten ist. Im Folgenden wird eine geschlossene Strategie vorausgesetzt. Der Heim-ISP beschreibt die für seine Kunden geltenden Autorisierungsregeln, indem er für sie die Zugriffsrechte festlegt. Dasselbe gilt für ein Unternehmen, welches für seine Mitarbeiter das gleiche tut. Dazu legt der Heim-ISP eine Datenbank an, welche der Partei, die dafür verantwortlich ist die Autorisierungsentscheidung, ob ein Paar (s, p) eine Beziehung $S \times P$ ist, zu treffen, wobei s ein Subjekt aus der Menge der Subjekte S und p ein Zugriffsrecht aus der Menge der Zugriffsrechte P ist. Ein Zugriffsrecht $p = (o, a)$ beschreibt eine erlaubte Aktivität a für ein Objekt o. Der Heim-ISP übertragt die für eine Autorisierungsentscheidung notwendigen Daten auf passende Weise zum RSP. Die Daten könnten z. B. als ACL bereitgestellt werden oder dem RBAC Ansatz folgend, so dass

entweder explizit die $S \times P$ Beziehungen angegeben werden oder Beziehungen aus denen diese vom RSP abgeleitet werden können.

Wenn ein Nutzer eine Verbindung bei einem kontaktierten ISP herstellen möchte, dann authentifizieren sich Nutzer und ISP gegenseitig. Nach einer positiven Validierung der vom Nutzer angegebenen Identität, sendet der kontaktierte ISP eine Autorisierungsanfrage an den RSP. Diese Anfrage enthält zusätzlich das Authentifikationsergebnis, welches dem RSP erlaubt zu überprüfen, ob der kontaktierte ISP die Identität des Nutzers wirklich verifiziert hat. Dieses Authentifizierungsergebnis enthält eine Signatur des Nutzers, die während des zwischen Nutzer und kontaktiertem ISP abgewickelten Authentifizierungsprotokolls ausgetauscht wurde. Wenn der Verdacht besteht, dass der ISP den Nutzer nicht korrekt authentifiziert hat, kann der RSP die vom Nutzer erklärte Identität überprüfen, damit er die Autorisierungsentscheidung auf Basis der korrekten Identität des Nutzers treffen kann. Auf einem Tripel (s, o, a) basierend, fällt der RSP die Autorisierungsentscheidung. Entsprechend seiner Entscheidung antwortet der RSP dem kontaktierten ISP mit “ja” oder “nein”. Entsprechend dieser Antwort setzt der kontaktierte ISP dann die Autorisierungsentscheidung durch. Abhängig von der Durchsetzung des ISP, kann der Nutzer dann ein Objekt nutzen oder nicht.

Diese Architektur wurde aus den folgenden Gründen gewählt:

- **Unterstützung von MIP:** Wenn Autorisierungsentscheidungen von einer anderen Partei als dem kontaktierten ISP getroffen werden soll, wird beim Einsatz von MIP ein Autorisierungsprotokoll benötigt, welches das „Pull“-Modell verwendet. Das „Push“- und das „Agent“-Modell unterstützen MIP nicht [VCFG00]. Da die hier vorgeschlagene Lösung wie in Kapitel 2 ausgeführt dem Einsatz von MIP gerecht werden soll, ist das „Pull“-Modell die einzig sinnvolle Lösung.
- **Aktualität der Autorisierungsregeln:** Da Informationen, auf denen die Autorisierungsentscheidung beruht immer aktuell sein sollten, muss es möglich sein die entsprechenden Daten schnell zu aktualisieren. Aus diesem Grund wird in der hier vorgestellten Lösung eine zentrale vom RSP verwaltete Datenbank eingesetzt, welche alle notwendigen Informationen enthält. Andere Ansätze, bei denen diese Daten verteilt sind, benötigen mehr Zeit für die Aktualisierung der Daten. Ein Beispiel hierfür sind als Ergänzung zu den Nutzerzertifikaten Attributzertifikate, welche Informationen über Zugriffsrechte oder Rollen enthalten, wie in [ChOt02] vorgeschlagen. Selbst wenn diese Zertifikate schnell zurückgerufen werden können, macht dieser Ansatz die Erzeugung und Verteilung neuer Zertifikate notwendig, was im Vergleich zur einfachen Änderung eines Datenbankeintrages ein langwieriger und teurer Prozess ist. Wenn von ISPs oder Unternehmen die Zugriffsrechte von größeren Mengen von Nutzern geändert werden, oder die Rollen von vielen Nutzern geändert werden, dann sind Zugriffsrechte in Zertifikaten nicht sinnvoll. Je öfter die Rechte der Nutzer geändert werden müssen, desto nachteilhafter ist dabei der Einsatz der Zertifikate.
- **Unterstützung der Flexibilität der Autorisierungsregeln:** Für den Fall das Autorisierungsregeln häufig geändert werden müssen, ist ein auf einer zentralen

Datenbank basierender Ansatz besser geeignet, als der Ansatz des Einsatzes von Zertifikaten, welche Zugriffsrechte oder Rollen enthalten. Wenn die Autorisierungsregeln häufig geändert werden, wachsen CRLs sehr schnell an und der durch die Zertifikate entstehende Verwaltungsaufwand wird unverhältnismäßig groß.

- **Privatsphäre und Geheimhaltung:** Die Tatsache, dass Rollen und Zugriffsrechte dem kontaktierten ISP nicht völlig offen gelegt werden, trägt dem Recht des Nutzers auf Wahrung seiner Privatsphäre Rechnung. Der kontaktierte ISP erhält nur über die Zugriffsrechte der vom Nutzer angefragten Aktivitäten und Ressourcen Auskunft, wenn die „ja"- bzw. „nein"-Antwort vom RSP erhält. Alle anderen Zugriffsrechte bleiben dem kontaktierten ISP verborgen. Die ISP könnte zusätzliche Anfragen an den RSP stellen, um die Rechte des Nutzers auszuspionieren. Dies kann konterkariert werden, indem eine Signatur des Anfragenden Nutzers und das Anfragedatum mit Zeitwert in die Anfrage integriert wird, wie bereits in der Authentifizierungslösung beschrieben. Diese Signatur sollte innerhalb der Autorisierungsanfrage an den RSP geschickt und von diesem überprüft werden. Wenn der RSP Anfragen ohne Signatur des Nutzers mit einem passenden Zeitstempel nicht beantwortet, dann kann der ISP nicht mehr Informationen erlangen, als er haben muss.

Damit ist erklärt, wie die eingangs gestellten Fordrungen bezüglich „Privatsphäre und Geheimhaltung", „Unterstützung der Flexibilität der Autorisierungsregeln" und „Aktualität der Autorisierungsregeln" erfüllt sind. Die eingangs unter dem Punkt „Protokoll" gestellte Anforderung und die „Sicherheit" können mit SAML, wie in Kapitel 7.2.2 beschrieben, erfüllt werden. Die unter „Verfall und Widerruf" sowie „Effizienz" aufgestellten Forderungen können durch den Einsatz von RBAC, wie in Kapitel 7.2.4 beschrieben, erreicht werden.

Wenn dem RSP das Ergebnis der Authentifizierung des Nutzers in der Autorisierungsanfrage präsentiert wird, kann er entscheiden, ob er dieses noch mal überprüft oder nicht. Da der RSP mit vielen ISPs in Verbindung steht ist es zu erwarten, dass er dementsprechend viele Autorisierungsanfragen zur selben Zeit zu bearbeiten hat. Die Verifizierung der Signaturen könnte je nach benötigten kryptographischen Verfahren und Mechanismen zeitaufwendig sein. Der hierdurch möglicherweise entstehende Engpass, könnte dazu führen, dass der RSP nicht alle ich ihm präsentierten Ergebnisse überprüfen will. Er könnte auch manche gleich und manche zu einem späteren Zeitpunkt verifizieren wollen. Bezüglich der Überprüfung der Signaturen bieten sich hier für den RSP unterschiedliche Strategien an:

- Der RSP verifiziert die Authentifizierungsergebnisse in Abhängigkeit vom Wert der angefragten Ressource.
- Der RSP verifiziert die Authentifizierungsergebnisse vom Vertrauen, dass er in den kontaktierten ISP hat, der ihm das Ergebnis schickt.
- Der RSP verifiziert die Authentifizierungsergebnisse mit einer bestimmten Wahrscheinlichkeit, so wie es bei einigen „Micropayment"-Systemen mit wahrscheinlichkeitsbasierter Verifikation der Fall ist [GaSi96, JaOd97].

Wenn der RSP nicht jedes an ihn geschickte Authentifizierungsergebnis überprüft, liegt der Fall einer wahrscheinlichkeitsbasierten Überprüfung vor. In diesem Fall darf der kontaktierte ISP auf keine Fall in der Lage sein vorauszusagen, ob der RSP die an Ihn geschickten Daten überprüft oder nicht.

Stand der Technik wäre für die Autorisierung ebenso wie für die Authentifizierung Diameter einzusetzen. Auf Grund von den im nächsten Abschnitt beschriebenen Hindernissen, wird hier eine auf dem Abschnitt 7.2.2 beschriebene SAML basierende Autorisierungslösung vorgeschlagen.

7.2.1 Hindernisse bei Diameter für die Autorisierung

Diameter kombiniert normalerweise Authentifizierung und Autorisierung. Das Basisprotokoll sowie die Erweiterungen MIPv4 und NASREQ erlauben nur eine Kette von Anfragen und Antworten, wobei jeweils die folgenden Anfragen und Antworten von den vorhergehenden abhängen. Gegenwärtig erlaubt Diameter nicht Authentifizierungsinformation gleichzeitig mit einer Autorisierungsanfrage vom kontaktierten ISP zum RSP weiterzuleiten. Um dies zu erreichen, müssten neue AVPs definiert werden, die innerhalb von Diameter verwendet werden können.
Ein weiterer Nachteil des Einsatzes von Diameter ist, dass die gegenwärtige Version von Diameter eine spezifische Menge an Protokollen für die Interaktion voraussetzt. Wenn SAML für Authentifizierung und Autorisierung verwendet wird, entfallen derartige Restriktionen.

7.2.2 SAML

Die Security Assertion Markup Language (SAML) ist ein XML-basiertes „framework“ für den Austausch sicherheitsrelevanter Information, wie z.B. für die Authentifizierung und Autorisierung. Die wesentlichen Vorteile von SAML für den Einsatz im Rahmen einer Architektur für „Sicheres Übergangsloses Roaming“ sind:

- Single Sign On: Das bedeutet, dass die Authentifizierungs- oder Autorisierungsinformation eines Nutzers von einer Anwendung zu einer anderen ohne die Notwendigkeit der Nutzerinteraktion weitergegeben werden kann.
- Interoperabilität: Das bedeutet, dass RSP, ISPs und Unternehmen sicher Authentifizierungs- und Autorisierungsinformationen austauschen können. SAML dient dazu, dass unterschiedliche Systeme, auf Sicherheit bezogene Daten austauschen.
- Offene Lösung: SAML wurde für die Zusammenarbeit mit vielen Standardtransportprotokollen entwickelt.

Die ausgetauschte Information wird in Form von so genannten Zusicherungen – „assertions“ – über Subjekte ausgetauscht. Subjekte werden in SAML als Entitäten, die verschiedene Identitäten in Sicherheitsbereichen besitzen, verstanden, wie z. B. das Subjekt Alice mit ihrer Passwortidentifikation. Die grundlegenden SAML Komponenten sind die “assertions”. Eine “assertion” ist ein von einer SAML Autorität in Erwartung eines Authentifikationsvorganges erstellter Datensatz, eine Attributinformation über ein

Subjekt oder eine Entscheidung darüber, ob ein Subjekt auf eine bestimmte Ressource zugreifen darf oder nicht. Eine solche “assertion” beinhaltet eine oder mehrere Stellungnahmen - “Statements”, welche vom Aussteller gemacht wurden. SAML erlaubt Ausstellern drei verschiedene Arten von „Zusicherungs-Stellungnahmen“:

- Authentifizierung: Das spezifizierte Subjekt wurde für einen bestimmten Zweck zu einer bestimmten Zeit authentifiziert.
- Autorisierungs-Entscheidung: Die Anfrage eines Subjektes auf ein bestimmtes Objekt zugreifen zu dürfen, wird gewährt oder abgelehnt.
- Attribut: Das spezifische Subjekt ist assoziiert mit den gegebenen Attributen.

SAML-„Assertions“ haben eine verschachtelte Struktur. Eine Reihe innerer Elemente repräsentieren Authentifikations-„Statements“, Autorisationsentscheidungs-„Statements“ oder Attribut-„Statements“. Sie enthalten entsprechende einschlägige Informationen, während ein äußeres generisches Zusicherungselement Informationen in sich trägt, die allen Zusicherungen gemein sind.

Das SAML Protokoll unterstützt “push” und “pull” von “data assertions” einer autorisierenden Quelle zu einem Empfänger. Es definiert ein “request/response-” Protokoll, bei welchem Clients Zusicherungen von SAML „Authorities“ erfragen können und Antworten von diesen erhalten (SAMLQuery and SAMLQueryResponse). Es sind drei Arten von “Authorities” definiert, nämlich “authentication authorities”, “attribute authorities” und “policy decision points”. Neben externen Datenbanken, die Strategien enthalten, können diese Autoritäten auch „Assertions“, welche sie in an sie gerichtete Anfragen erhalten haben, für die Erzeugung von „Assertions“ nutzen. SAML Autoritäten benutzen „Assertions“ auch bei der Erzeugung neuer „Assertions“. Policy decision points (PDP) sind Autoritäten, die Autorisierungsentscheidungen für sich selbst oder andere Entitäten, die nach einer solchen Entscheidung fragen, treffen.

SAML erlaubt es außerdem “Assertions” über Standard-Internet-Protokolle zu verteilen, indem es SAML Information an Transport und Nachrichten framework Standards bindet. jede Bindung wird ein Name gegeben nach dem Muster „SAML x Binding“. SAML definiert gegenwärtig nur eine Bindung, nämlich SOAP über http, wie in Abbildung 142 dargestellt.

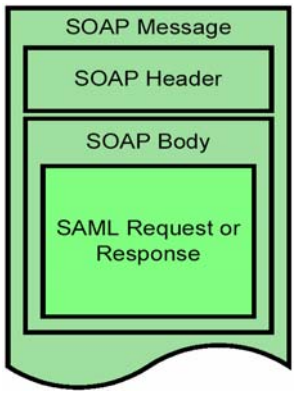


Abbildung 142: SAML Kapselung in SOAP

7.2.3 Autorisierungslösung

Es wird im Folgenden vorausgesetzt, dass der kontaktierte ISP den Nutzer schon authentifiziert hat. Er leitet die Authentifikationsinformation zum RSP weiter. Nach der Authentifizierung des Nutzers, sendet der ISP eine Autorisierungsentscheidungsanfrage – ein SAML-“Authorization Decision Query”, welche in einer „Assertion“ Autorisierungsinformation enthält, zum RSP. Der RSP beantwortet dies mit einer SAML-“Response”. Auf die grafische Darstellung dieses Protokolls wird wegen seiner Einfachheit hier verzichtet.

SAML Nachrichten – Anfragen und Antworten - beinhalten für gewöhnlich Zusicherungen – SAML-„Assertions“. Eine SAML-“Assertion” ist eine Erklärung einer zu irgendjemand zugeordneten Tatsache. So eine Zusicherung kann digital signiert werden. Es ist möglich SAML mit speziellen Arten von „Assertions“ und „Statements“ zu erweitern.

Das Beispiel eines SAML-“Authorization Decision Query” zeigt Abbildung 143. Dieses Beispiel zeigt eine SAML “Request” Nachricht, die eine SAML-“Authorization Decision Query” darüber enthält, ob es einem Subjekt erlaubt ist, eine IPSec VPN Verbindung zu etablieren. Die angegebene Identität kann durch Auswertung der Daten, die das “Assertion” Element enthält, überprüft werden.

```
<samlp:Request …
 MajorVersion="1"
 MinorVersion="0"
 IssueInstant="UTC Date / Time"
 RequestID="128.14.234.20.12345678"
 RespondWith="AuthorizationDecisionStatement">
     <samlp:AuthorizationDecisionQuery
          Resource="IPSec VPN connection">
          <saml:Subject>
               <saml:NameIdentifier
                SecurityDomain="smithco.com"
                Name="joeuser" />
          </saml:Subject>
          <saml:Action>connect</saml:Action>
          <saml:Evidence>
               <saml:Assertion>…</saml:Assertion>
          </saml:Evidence>
     </samlp:AuthorizationDecisionQuery>
</samlp:Request>
```

Abbildung 143: SAML Abfrage einer Autorisationsentscheidung

Ein “Authorization Decision Query” enthält ein Element, welches als “Evidence” bezeichnet wird. Eine “Evidence” ist eine Liste von “Assertions”. Zusätzliche

“Assertion”-Elemente, welche benötigte Authentifikationsinformation enthalten können, können auch Bestandteil eines „Authorization Decision Query“ sein. Eine “Assertion”, in der notwendige Authentifikationsinformation gespeichert ist, muss Bestandteil des „Authorization Decision Query“ sein, welches vom ISP gesendet wird. SAML definiert einen “Assertion” Typ, der als “Authentication Statement” dient, welcher in diesem Fall genutzt werden sollte. Das Beispiel eines solchen SAML-“Authentication Statement” zeigt die folgende Abbildung 144.

```
<saml:Assertion_…>
     <saml:AuthenticationStatement
          AuthenticationMethod=“…URI…”
          AuthenticationInstant=“2001-12-03T10:02:00Z”>
          <saml:Subject>
               <saml:NameIdentifier
                SecurityDomain=“smithco.com”
                Name=“joeuser” />
               <saml:SubjectConfirmation>
                    <saml:ConfirmationMethod>
                     …URI…
                    </saml:ConfirmationMethod>
                    <saml:SubjectConfirmationData>
                     …authentication information…
                    </saml:SubjectConfirmationData>
               </saml:SubjectConfirmation>
          </saml:Subject>
     </saml:AuthenticationStatement>
</saml:Assertion>
```

Abbildung 144: Beispiel eines “Authentication Statement”

```
<saml:Assertion_…>
    <saml:AttributeStatement>
     <saml:Subject>…</saml:Subject>
     <saml:Attribute
      AttributeName="Location"
      AttributeNamespace="http://rsp.com">
          <saml:AttributeValue>
           somewhere
          </saml:AttributeValue>
     </saml:Attribute>
     <saml:Attribute
      AttributeName="some other attribute"
      AttributeNamespace="http://rsp.com">
          <saml:AttributeValue>
                some value
          </saml:AttributeValue>
     </saml:Attribute>
    </saml:AttributeStatement>
</saml:Assertion>
```

Abbildung 145: Beispiel eines "Attribute Statements"

Nachdem die Autorisierungsentscheidung getroffen wurde, antwortet der RSP mit einer „Response". Eine „Response" kann verschiedene Elemente des Typs „Assertion" beinhalten. Das Beispiel einer SAML-"Response" gibt die folgende Abbildung 146.

```
<samlp:Response
     MajorVersion="1" MinorVersion="0"
     IssueInstant="UTC Date / Time"
     RequestID="128.14.234.20.90123456"
     InResponseTo="128.14.234.20.12345678"
     StatusCode="Success">
     <saml:Assertion
          MajorVersion="1" MinorVersion="0"
          AssertionID="128.9.167.32.12345678"
          Issuer="RSP">
          <saml:Conditions
               NotBefore="2001-12-03T10:00:00Z"
               NotAfter="2001-12-03T10:05:00Z" />
          <saml:AuthorizationStatement …>
          …see 0
          </saml:AuthorizationStatement>
     </saml:Assertion>
</samlp:Response>
```

Abbildung 146: Beispiel einer "Response"

Eine Antwort auf eine Autorisierungsanfrage muss ein SAML-„Authorization Statement“ enthalten. Ein Beispiel dafür zeigt Abbildung 147.

```
<saml:Assertion_…>
     <saml:AuthorizationStatement
      Decision="Permit"
      Resource="IPSec VPN connection">
          <saml:Subject>…</saml:Subject>
          <saml:Action Namespace="http://rsp.com">
               connect
          </saml:Action>
     </saml:AuthorizationStatement>
</saml:Assertion>
```

Abbildung 147: Beispiel eines “Authorization Statement”

7.2.4 Autorisierungsentscheidung

Wenn der RSP die Autorisierungsanfrage vom kontaktierten ISP erhalten hat, muss er entscheiden, ob der anfragende Nutzer die Ressource benutzen darf oder nicht. Die Autorisierungsentscheidung erfordert für den RSP die Möglichkeit, die Identität des Nutzers zu nutzen. Der RSP kann entweder die Nutzer ID aus den Autorisierungsdaten extrahieren oder das Ergebnis der Authentifikation des kontaktierten ISPs nochmals überprüfen, bevor er die Autorisierungsentscheidung trifft.

Zusätzlich benötigt der RSP einige Information um zu wissen, welche Autorisierungsdaten verwendet werden müssen. Der RSP verfügt über Autorisierungsdaten von allen ISPs, mit denen er einen Vertrag abgeschlossen hat. Das Zertifikat des Nutzers sollte genug Information enthalten, um es dem RSP zu ermöglichen, die Autorisierungsentscheidung zu treffen. Das Zertifikat wird entweder im Rahmen des Authentifizierungsergebnisses zum RSP weitergeleitet oder im Delegationsfall dem RSP direkt präsentiert. Unter der Annahme, dass der Heim-ISP oder das Unternehmen unterschiedliche Mengen an Autorisierungsregeln beim RSP angegeben haben, kann die passende Menge von Regeln durch den Namen des ISPs oder Unternehmens ausgewählt werden, welcher im Zertifikat vorhanden sein muss. Dieses sollte leicht möglich sein, da beide, ISP und das Unternehmen, in den Zertifikatsausstellungsprozess involviert sind.

Der Heim-ISP und das Unternehmen müssen Ihre Autorisierungsdaten dem RSP so zu Verfügung stellen, dass er sie in passender Art und Weise anwenden kann. Es gibt mehrer verschiedene Ansätze für den RSP, um die für die Autorisierung der Subjekte bezüglich der Aktivitäten mit unterschiedlichen Objekten nötigen Informationen zu erlangen. Diese können einerseits auf der Verwendung von Zugangskontrolllisten (ACLs) oder Rollen basierter Zugriffskontrolle (RBAC) aufbauen. Diese unterschiedlichen Ansätze implizieren verschiedene Technologien, welche zum Erreichen der

Autorisierungsentscheidung eingesetzt werden müssen. Sie unterscheiden sich außerdem noch in den Verwaltungskosten.

Wenn ACLs eingesetzt werden, werden die Zugriffsrechte der Subjekte für die Objekte in Listen zusammen mit dem Objekt gespeichert. Für jedes Objekt gibt es ein Liste, die eine Kombination von Subjekten und den das spezifische Subjekt betreffende erlaubten Aktivitäten enthält. Diese Aktivitäten können mit zusätzlichen Bedingungen kombiniert werden, wie z.B. der Zeit in der ein Subjekt auf das Objekt zugreifen darf.

Abbildung 148 zeigt das Beispiel einer ACL. Gemäß den Einträgen dieser ACL darf der Nutzer Alice als Subjekt IPSec VPN Verbindungen und MIP Verbindungen zwischen 0 a.m. und 12 p.m. nutzen, wohingegen Nutzer Bob nur die Erlaubnis hat, IPSec VPN Verbindungen zwischen 8 a.m. und 6 p.m. aufzubauen.

Objects	Subjects and their permitted activities
IPSec VPN connection	Alice: connect (0 a.m. < t < 12 p.m.); Bob: connect (8 a.m. < t < 6 p.m.)
MIP connection	Alice: connect (0 a.m. < t < 12 p.m.)

Abbildung 148: Beispiel einer ACL

Wenn Nutzer Bob eine IPSec VPN Verbindung beim kontaktierten ISP anfordert, dann sollte der kontaktierte ISP die folgenden Parameter für seine SAML „Autorisation Decision Query“ bereithalten, um sie zum RSP zu senden: $s =$ ”Bob”, $o =$ ”IPSec VPN connection”, $a =$ ”connect”. Eine Zeitangabe ist nicht notwendig, da der RSP die Zeit zu der er die Anfrage erhalten kann für die Autorisierungsentscheidung benutzen kann. Wenn der RSP z. B. eine Anfrage um 3 p.m. erhält, dann schaut er in der Datenbank mit den Autorisierungsregeln nach, um die Autorisierungsentscheidung zu treffen und generiert eine SAML „authorization response“ mit der Autorisierung „*yes*“.

Wenn exklusiv eine einzelne Aktivität in einem spezifischen Autorisierungs-Kontext vorgeschlagen wird, wie im Beispiel von Abbildung 148, dann kann die Aktivität in der ACL vermieden werden. In Abhängigkeit von der unterstützten Granularität der Objektdefinitionen, kann mehr oder weniger stark innerhalb der Menge der Objekte differenziert werden. Anstatt einfach „IPSec VPN Verbindung“ als Objekt zu betrachten, könnte man hinsichtlich des QoS Aspekts noch zwischen „IPSec VPN Verbindung ≤ 100 kbps“, „IPSec VPN Verbindung ≤ 500 kbps“ und „IPSec VPN Verbindung ≤ 1 Mbps“ unterscheiden. Entscheidungen über feiner granulare Objekte lassen sich auch mit Verwendung einer grobkörnigeren Objektstruktur in den Autorisierungsanfragen erreichen. So kann z.B. ein Nutzer, welcher „IPSec VPN Verbindung“ mit einem zusätzlichen Parameter wie z.B. einer maximalen Bandbreite „≤ 500 kbps“ nutzen darf, in der Autorisierungsanfrage nur „IPSec VPN Verbindung“ angeben. Der RSP gibt in seiner Antwort dann zusätzlich zum „ja“ – vorausgesetzt der Nutzer ist autorisiert – den einschränkenden Parameter „≤ 500 kbps“ als zusätzliche Bedingung mit.

Die Verwendung von ACLs hat ein paar Nachteile bezüglich der Administration. Das Verwalten von Zugriffsrechten spezieller Nutzer kann ziemlich kostspielig sein. Daher gibt es andere Ansätze wie z.B. RBAC, die eine effizientere Administration erlauben. Die Möglichkeit effizient Nutzerdaten zu verwalten ist wichtig, besonders dann, wenn ein ISP häufig die Zugriffsrechte seiner Nutzer ändert, um z.B. seine Wettbewerbsfähigkeit zu erhöhen.

Daher sollte für die Autorisierung hier auf das Konzept der rollenbasierten Zugriffskontrolle (RBAC) zurückgegriffen werden. Das grundlegende Konzept der RBAC ist bereits in Kapitel 7 erklärt. Im Folgenden wird mit Beispielen erklärt, wie es bei der zum Tragen kommt und welche Prinzipien beachtet werden müssen:

Wie in Kapitel 7 erwähnt, kennt RBAC den Mechanismus der Rollenaktivierung: Bei einer Rollenhierarchie erbt eine “Senior”-Rolle von einer “Junior”-Rolle. Beispiel für eine Vererbungshierarchie im Falle eines Unternehmens zeigt Abbildung 149. Die im Moment gültigen Zugriffsrechte eines Nutzers ergeben sich aber nicht notwendigerweise aus der Auswertung seiner untersten „Senior“-Rollen. Diese Rollen können ruhen. Die im Moment gültigen Zugriffsrechte ergeben sich aus den im Moment aktiven Rollen. Sitzungen definieren sich hier über Phasen, in denen ein Nutzer bestimmte Rollen annimmt. Die Sitzung eines Nutzers ist assoziiert mit einer oder mehreren Rollen. Ein Nutzer kann ein Unternehmensmitarbeiter sein oder als Privatanwender einfach nur im web surfen wollen. Er kann zwar beide Rollen gleichzeitig spielen, wenn die momentan aktive Rolle der Privatanwender ist, wird er allerdings keinen Zugriff auf Unternehmensressourcen haben.

Das Prinzip des “least privilege” sollte grundsätzlich beachtet werden. Dieses Prinzip des geringsten Privilegs besagt, dass der Nutzer nicht mehr Rechte erhalten sollte, als im Moment gerade notwendig. Wenn er also nur privat im Web surfen möchte, sollte er nicht gleichzeitig die Rechte eines Unternehmensmitarbeiters mit dem entsprechenden Zugriff auf Ressourcen des Unternehmens haben. Die Rolle des Unternehmensmitarbeiters sollte z. B. nicht gleichzeitig mit der des Privatanwenders aktiviert sein. Der Nutzer hat also verschiedene Zugriffsrechte zu verschiedenen Zeiten in Abhängigkeit von den gerade aktivierten Rollen. Konsequenter Weise müssen diese Rechte am Ende einer Sitzung widerrufen werden.

Wenn ein Nutzer gleichzeitig unterschiedliche Rollen spielt, kann es zu Konflikten hinsichtlich der aus den unterschiedlichen Rollen abgeleiteten Zugriffsrechten kommen, da eine bestimmte Aktion der einen Rolle erlaubt, gleichzeitig aber der anderen Rolle verboten sein könnte. Bedingungen zur „Trennung von Betriebsarten“ regeln Konflikte, die auftreten wenn sich die Zugriffsrechte von zwei Rollen, die dasselbe Subjekt innehat, einander widersprechen. Man unterscheidet hier statische und dynamische Trennung:

- Statische Trennung bedeutet, dass zwei Rollen oder Bedingungen, die sich gegenseitig ausschließen nicht zur selben Zeit demselben Subjekt zugeordnet werden dürfen. Das heißt wie schon oben gefordert, dass ein Nutzer nicht gleichzeitig Unternehmensmitarbeiter sein darf.

- Dynamische Trennung bedeutet, dass ein Subjekt zwar unterschiedliche Rollen innehaben kann, aber für eine einzelne spezifische Anforderung immer nur eine Rolle betrachtet wird. Das bedeutet, dass das Subjekt zwar unterschiedliche sich gegenseitig ausschließende Rollen innehaben kann, aber zu einer bestimmten Zeit für eine bestimmte Anforderung nur eine dieser Rollen aktiv sein kann. Der Nutzer ist also gleichzeitig Unternehmensmitarbeiter und Privatkunde eines ISPs. Wenn er jedoch auf die Ressourcen seines Unternehmens Zugreifen will, muss er als aktive Rolle die des Unternehmensmitarbeiters spielen. Er kann nicht gleichzeitig als Privatanwender frei surfen.

Bei RBAC wird Strategie-Neutralität vorausgesetzt. Das bedeutet, dass RBAC die Flexibilität bietet mit unterschiedlichen Sicherheitsstrategien zu Recht zu kommen. Es ist für die Autorisierungslösung wichtig ein Mittel zu haben, welches die Berücksichtigung einer großen Bandbreite unterschiedlicher Sicherheitsstrategien ermöglicht, da bei dem Szenario eines weltweiten Roamings eine Vielzahl unterschiedlicher Player beteiligt sein können, bei denen unterschiedlichste sich mit der Zeit ändernde Strategien zu erwarten sind. Da ISPs und Unternehmen eine große Anzahl an Kunden/Nutzern bzw. Mitarbeitern haben, stellen sich hier besondere Anforderungen an die Administration der Autorisierungsregeln. Des Weiteren können die ISPs daran interessiert sein, ihre Geschäftsmodelle zu verändern, entweder zum Zwecke der Optimierung oder um neue Ideen auszuprobieren, während Unternehmen daran interessiert sind, interne Umstrukturierungen vornehmen zu können. Dies impliziert normalerweise auch Änderungen in der Sicherheitsstrategie. Der Nutzer kommt in eine neue Kundenkategorie oder erhält eine neue Rolle innerhalb seines Unternehmens. Dementsprechend müssen die Zugriffsrechte angepasst werden. Dieser Tatsache kann mit der Verwendung von RBAC für die Verwaltung von Zugriffsrechten Rechnung getragen werden.

Ein Beispiel für eine Rollenhierarchie im Falle eines Unternehmens zeigt Abbildung 149. In diesem Beispiel gibt es einen „department manager“, einen „sales employee“, einen „homeworking employee“ und einen „ordinary employee“. Wie üblich hat der “department manager” die höchsten Privilegien. Er erbt die Zugriffsrechte aller anderen Rollen und hat zusätzlich noch andere Zugriffsrechte. Der “sales employee” ist üblicherweise mobil, um die Produkte des Unternehmens potentiellen Kunden zu präsentieren. Er benötig daher das Recht zu den normalen Geschäftszeiten eine MIP Verbindung zu nutzen. Der “homeworking employee” benötigt die Möglichkeit von zu Hause Daten zum Untenehmen zu übertragen und umgekehrt. Daher hat er 24 Stunden am Tag das Recht eine IPSec VPN Verbindung zu seinem Unternehmen aufzubauen. Der “ordinary employee” hat das Recht abends eine IPSec VPN Verbindung zu seinem Unternehmen aufzubauen, um z.B. tagsüber liegen gebliebene Arbeit von zu Hause noch zu erledigen. In der Realität können die Rollenhierarchien noch sehr viel komplexer sein, als indem hier gezeigten Beispiel. Wenn man anstatt eines Unternehmens einen ISP betrachtet, können die Rollen unterschiedliche Klassen von Kunden beschreiben, die auf Basis ihrer verschiedenen Verträge unterschiedliche Dienste nutzen.

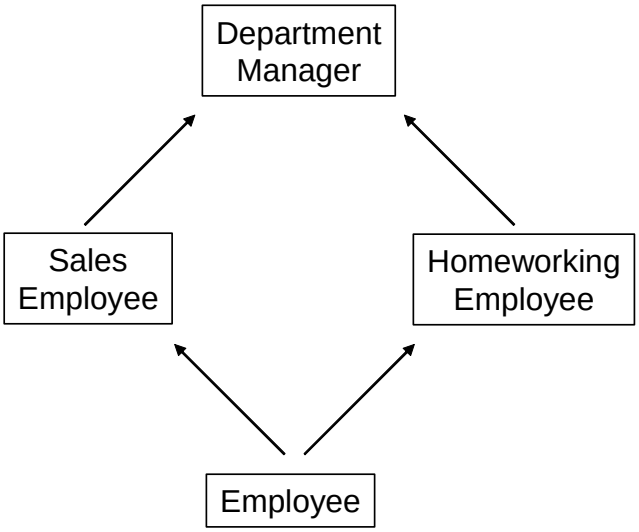


Abbildung 149: Beispiel einer Rollenhierarchie

Tabelle 7: Vererbungsbeziehungen von Rollen im Unternehmen.

Senior Role	**Junior Role(s)**
Department manager	Sales employee, Homeworking employee
Sales employee	Employee
Homeworking employee	Employee

Normalerweise ergibt sich die Rollenstruktur aus der Organisationsstruktur eines Unternehmens. Wenn der Prozess der Rollenerzeugung beendet ist, muss der Administrator den Mitarbeitern ihre Rollen zuordnen. Tabelle 8 zeigt Beispiele für Zuordnungen von Nutzern zu Rollen. Zu Beachten ist hierbei, dass einem einzelnen Nutzer mehrere Rollen zugeordnet sein können.

Tabelle 8: Beispiel einer Zuordnung von Rollen zu Nutzern in einem Unternehmen

User	**Role**
Alice	Department manager
Bob	Sales employee
Carol	Homeworking employee
David	Employee
…	…

Bei der Zuordnung von Rollen zu Zugriffsrechten, müssen nur die Zugriffsrechte eingeführt werden, die noch nicht in der Zuordnung der Zugriffsrechte zu den Junior Rollen enthalten sind, da Zugriffsrechte der Junior-Rollen automatisch geerbt werden.

Gemäß der in Tabelle 9 abgebildete Zuordnung von Zugriffsrechten zu Rollen z. B. hat ein „department manager“ Das Recht MIP und IPSec Verbindungen den ganzen Tag über aufzubauen. Das Recht für den Aufbau der IPSec VPN Verbindung erbt er ja von der Rolle des „homeworking employee“.

Tabelle 9: Beispiel einer Zuordnung von Zugriffsrechten zu Rollen.

Role	**Permission**
Department manager	MIP connection (0 a.m. – 12 p.m.)
Sales employee	MIP connection (8 a.m. – 10 p.m.)
Homeworking employee	IPSec VPN connection (0 a.m. – 12 p.m.)
Employee	IPSec VPN connection (6 p.m. – 10 p.m.)

Bei dem in Tabelle 9 dargestellten Beispiel ist das Zeitintervall für den “sales employee” vollständig in dem angegebenen Zeitintervall für den “department manager” enthalten. Dasselbe gilt auch für die Zeitintervalle, in denen der “ordinary employee” und der „homeworking employee“ Zugriffsrechte haben. Auf Grund der Vererbungsbeziehungen wäre es auch möglich für den “department manager” Zeitintervalle von 0 a.m. – 8 a.m. für MIP Verbindungen und 10 p.m. – 12 p.m. für IPSec VPN Verbindungen festzulegen ohne sein resultierendes Zugriffsrechte zu verändern.

Wenn der Heim ISP seine Strategie zum RSP übertragen hat und der RSP seinen RBAC „engine“ installiert hat, dann ist der Nachrichtenaustausch zwischen dem Heim - ISP und dem RSP genauso, wie bei der Verwendung von ACLs. Wenn ein Nutzer eine Verbindung vom kontaktierten ISP aus herstellen will, dann sendet der kontaktierte ISP die relevanten Daten s, o, a zum RSP innerhalb der SAML Autorisierungsentscheidungs-Query. Um die Autorisierungsentscheidung zu treffen, extrahiert der RSP die Nutzer ID aus dem Query. Die Rolle ermittelt er aus der ihm bekannten Zuordnungstabelle, die er vom ISP erhalten hat. Danach prüft er noch, welche anderen Rollen der Nutzer der Rollenhierarchie entsprechend noch spielt. Im letzten Schritt der Autorisierungsentscheidung prüft der RSP, ob der Nutzer entsprechend der Rollen die er innehat, das Recht hat, die angefragte Ressource zu nutzen. Das Ergebnis dieser Entscheidung wird dann als „ja“ oder „nein“ eventuell mit zusätzlichen Bedingungen an den kontaktierten ISP weitergegeben, damit dieser dann die Autorisierungsentscheidung durchsetzen kann.

7.3 Lösung für Hochgeschwindigkeits-Anforderung

Die im vorigen Abschnitt 7.2 dargestellte Lösung erfüllt zunächst die Anforderungen aus Abschnitt 7.1. Im Folgenden wird eine Lösung dargestellt, mit welcher hauptsächlich der Anforderung der Hochgeschwindigkeit Rechnung getragen werden soll. Die bisher gestellten Anforderungen werden immer noch berücksichtigt und zusätzlich noch die Anforderung an die Geschwindigkeit des Autorisierungsprozesses.

Nachdem der Nutzer eine Anfrage nach einem Dienst gestellt hat, ist es wünschenswert, dass der kontaktierte ISP die Autorisierungsentscheidung schnellst möglich treffen kann. Um den Autorisierungsprozess zu beschleunigen, kann der kontaktierte ISP versuchen, die Autorisierung eigenständig ohne Unterstützung durch den ISP durchzuführen. Wenn das möglich ist, findet keinerlei Austausch von SAML-Nachrichten statt. Falls der kontaktierte ISP in der Lage ist, ohne Hilfe des RSPs die Autorisierungsentscheidung zu treffen, tritt eine Verkürzung der Dauer des Autorisierungsvorgangs ein.

Die Hochgeschwindigkeitslösung basiert auf der Verwendung von zu X.509 konformen Zertifikaten mit neu definierten Erweiterungen für Autorisierungszwecke. Der Autorisierungsprozess ist schematisch in Abbildung 150 dargestellt. Mehr Details über X.509 ID-Zertifikate können in [HPFS02] gefunden werden. Konzeptionell wird hier eine Lösung vorgeschlagen, die dem Pull Modell, wie unter [VCFG00] beschrieben ähnelt. Die hier vorgeschlagene Lösung erfordert jedoch nicht, dass der mobile Nutzer eine Verbindung zu einem AAA Server herstellen muss jedes Mal, wenn er eine Anfrage zu einem kontaktierten ISP schickt, um die notwendigen Autorisierung zu erhalten. Stattdessen wird vorausgesetzt, dass der Nutzer über ein X.509 Zertifikat verfügt, das die Autorisierungsdaten beinhaltet, welche der kontaktierte ISP benötig, um seine Entscheidung über die Autorisierung des Nutzers zu treffen. Das entsprechende Zertifikat wird von der CA ausgestellt und nicht von einem AAA Server wie ein Ticket. Dementsprechend ist die Gültigkeitsdauer des Zertifikates sehr viel länger, als die eines von einem AAA Server ausgestellten Tickets.

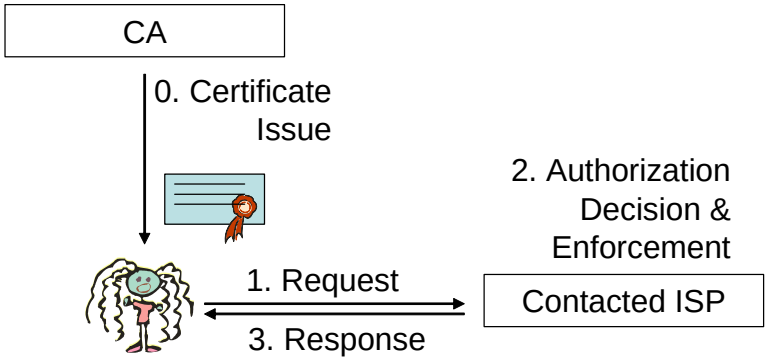


Abbildung 150: Autorisierungsprozess basierend auf Zertifikaten

Die CA stellt das Zertifikat bereits mit den für die Autorisierung notwendigen Erweiterungen aus. Dies ist möglich, da in dem zugrunde gelegten Geschäftsmodell die CA entweder dem RSP oder dem Unternehmen, bei dem der Nutzer beschäftigt ist assoziiert ist.

Wenn das Zertifikat von einer dem RSP assoziierten CA ausgestellt wurde, registriert sich der Nutzer bei seinem Heim-ISP für eine bestimmte Zeit. Dieser Heim-ISP spielt die Rolle der RA. Der Heim-ISP leitet als RA alle relevante Information an die CA weiter. Diese Daten beinhalten einige Informationen bezüglich der mobilen

Kommunikationsdienste für die der Nutzer autorisiert ist. Dann generiert die CA das X.509 Identitäts-Zertifikat für den Nutzer, welches spezifische Erweiterungen für die Autorisierung von Diensten im Speziellen für den Roaming Dienst enthält.

Im Falle der Unternehmens-CA generiert das Unternehmen die Autorisierungen für seine Mitarbeiter bezüglich der Roaming Dienste.

Offensichtlich ist es in beiden Fällen der Aussteller CA möglich, die X.509 Zertifikate zu generieren, welche die relevanten Autorisierungsinformationen enthalten. Dies ist möglich, da die CA der Partei, welche die Autorisierung, die die Nutzung der mobilen Dienste betrifft, definiert, assoziiert ist. Dieselben Zertifikate werden hierbei sowohl für die Authentifikation als auch die Autorisierung genutzt. Sie beinhalten sowohl die ID des Nutzers, als auch Informationen darüber, welche mobilen Kommunikations-Dienste er nutzen darf im Kontext des Roaming. Wenn der mobile Nutzer einen Dienst nur für eine begrenzte Zeit nutzen möchte, impliziert dies hier, dass auch die Gültigkeit des Zertifikates entsprechend zeitlich begrenzt sein muss. An dieser Stelle sei angemerkt, dass ein Nutzer mehrere X.509v3 ID Zertifikate gleichzeitig nutzen kann.

In der hier entwickelten Lösung werden Zertifikatserweiterungen definiert, um den kontaktierten ISP mit der bei der Autorisierung notwendigen Information zu versorgen. Die Erweiterungen sind so beschaffen, dass alle relevanten Aspekte für Regeln für den Gebrauch von mobilen Kommunikationsdiensten beachtet werden.

Die zu schützenden Ressourcen, um die es hier geht, betreffen im Wesentlichen den Nutzen mobiler Kommunikationsdienste und nicht die generellen Kapazitäten des Nutzers. Daher sind die Erweiterungen für die Zertifikate so beschaffen, dass sie vor allem diese Aspekte abdecken. Es ist trotzdem möglich zusätzliche hier nicht aufgeführte andere Erweiterungen hinzuzufügen. Bei Unternehmens-Zertifikaten ist es sinnvoll andere neue Erweiterungen in die Zertifikate zu integrieren, welche unternehmensinterne Zugriffsrechte regeln. Allgemein betrachtet stellen Zertifikatserweiterungen, wie sie in X.509v3 definiert sind, Methoden zur Assoziation zusätzlicher den Nutzer betreffender Attribute dar. Einige Erweiterungen werden bereits in [HPFS02] beschrieben. Die X.509v3 Zertifikatsformate erlauben es eigene Erweiterungen für bestimmte Gemeinschaften und Zwecke zu definieren, welche prinzipiell beliebige Informationen, die nur für diese geschlossenen Gemeinschaften relevant sind, enthalten können.

Der Grund dafür Berechtigungen von Nutzern in die Zertifikatserweiterungen zu schreiben anstatt zusätzliche Attributszertifikate zu verwenden ergibt sich aus der Anforderung der Hochgeschwindigkeit. Die Autorisierungsentscheidungen sollten so schnell wie möglich verfügbar sein. Ein zusätzliches Attributszertifikat würde eine weitere Überprüfung eines Zertifikates – des zusätzlichen Attributszertifikates nämlich - bedeuten. Dies beinhaltet zusätzlich die Verifikation einer digitalen Signatur und einen weitere Überprüfung dessen, ob das Zertifikat zurückgerufen wurde oder nicht. Daher sollte die Korrektheit der erklärten Nutzeridentität und der Berechtigungen des Nutzers mit einer einzigen Signatur zu überprüft werden können. Des Weiteren sollte die Menge

der Zertifikatsdaten, welche überprüft und übertragen werden müssen kompakt sein. Die vorgeschlagene Lösung ist so generisch, dass sie auch auf zukünftige andere noch nicht spezifizierte Geschäftsmodelle angewandt werden können.

Es wird hier eine geschlossene Strategie vorausgesetzt. Das bedeutet, dass alles, was nicht ausdrücklich erlaubt ist, verboten ist.

Die im folgenden dargestellte Struktur für Erweiterungen definiert mittels der Berechtigungen klar die Zugriffsrechte des Zertifikatseigentümers bezüglich spezifischer Objekte in Kombination mit spezifischen Nutzungsbedingungen.

Gegenwärtig ist die hier beschriebene Struktur für die Erweiterungen eher eine Möglichkeit für die Beschreibung einer Ressource. Sie kann erweitert werden, um eventuell andere Aspekte der Nutzung von Roaming Diensten zu berücksichtigen.

Entsprechend dem X.509v3 Standard, welcher in [HPFS02] beschrieben ist, werden die Erweiterungen auf die im Folgenden dargestellte Weise definiert. Die zu signierenden Zertifikatserweiterungen werden unter Beachtung der ASN.1 Kodierungsregeln „distinguished encoding rules“ (DER) erstellt. Im einzelnen gilt:

```
Extension ::= SEQUENCE  {
     extnID      OBJECT IDENTIFIER,
     critical    BOOLEAN DEFAULT FALSE,
     extnValue   RSExtension
}
```

Die oben beschriebene Erweiterung ist nicht kritisch. Dies ist darin begründet, dass eine Anwendung, die diese spezifische Erweiterung nicht erkennt – das Zertifikat kann noch in einem anderen Kontext außerhalb der Welt der mobilen Dienste verwendet werden - , das Zertifikat aufgrund dessen nicht zurückweisen wird. Dies unterstützt die Einsetzbarkeit solcher Zertifikate. Sie können beispielsweise auch für die Authentifikation der Nutzer im Rahmen von Anwendungen innerhalb von diversen Geschäftsprozessen in einem Unternehmen eingesetzt werden. Da ein mobiler Nutzer die Erlaubnis haben kann verschiedene Roaming Dienste zu nutzen, kann diese Erweiterung eine entsprechende Menge von Diensten enthalten. Außerdem kann die Erweiterung einige weitere Nutzungsbedingungen enthalten, so wie ein Maximum für QoS, maximales Datenvolumen. Maximale Kosten in einer angegebenen Währung, die ein Nutzer verursachen darf. Diese Nutzungsbedingungen gelten für alle vom Nutzer in Anspruch genommenen Dienste. Es gilt:

```
RSExtension ::= SEQUENCE  {
     roamingServices  RoamingServices,
     qos              INTEGER,
     volume           INTEGER,
     maxCosts         INTEGER,
```

```
     currency         Currency
}

Currency ::= ENUMERATED  {
     EUR              (0),
     USD              (1),
     YEN              (2)
}

RoamingServices ::= SET OF RoamingService
```

Zu jedem Roaming Dienst gibt es wiederum eine Menge von Diensten, welche aus einer Folge von Elementen besteht welche die erlaubten Dienste, die Eigenschaften dieser Dienste und einige entsprechende Nutzungsbedingungen für spezielle Nutzer beschreiben. Diese beinhalten die Spezifikationen der Tage, an denen ein Dienst genutzt werden kann, z.B. Montag bis Freitag, die Tageszeit, z.B. 8:00 Uhr Bis 19:00 Uhr, und die Zieladressen, zu denen einer Nutzer Verbindung aufnehmen darf:

```
RoamingService ::= SEQUENCE  {
     permService      PermService,
     days             Days,
     daytime          Daytime,
     permDestinations PermDestinations
}

Days ::= SET OF Day

Day ::= ENUMERATED  {
     Monday           (0),
     Tuesday          (1),
     Wednesday        (2),
     Thursday         (3),
     Friday           (4),
     Saturday         (5),
     Sunday           (6)
}

Daytime ::= SEQUENCE  {
     notBeforeHour    Time,
     notAfterHour     Time
}
```

Im Kontext des hier zugrunde gelegten Geschäftsmodells kann ein erlaubter Kommunikationsdienst entweder aus einem einfachen Internet Roaming Zugangs Dienst, einem Dienst der VPN unterstützt, einem Dienst, der mit Mobile IP übergangsloses

Roaming gewährleistet, und einem Dienst zur Unterstützung von VPN über Mobile IP bestehen. Im Allgemeinen ist diese Liste von Diensten erweiterbar und sie kann angepasst werden um weitere Dienste zu unterstützen.

```
PermService ::= ENUMERATED  {
     INTERNET         (0),
     VPN              (1),
     MIP              (2),
     MIP-VPN          (3)
}
```

Die Erlaubnis für einen Roaming Dienst kann von der Zieladresse abhängen. Daher wird eine Zieladresse als Nutzungsbedingung für einen erlaubten Dienst angesehen. Die Menge der Ziele kann mehrere erlaubte Zieladressen umfassen. Diese Menge von Adressen wird als eine Menge erlaubter und eine Menge verbotener Ziele beschrieben. Die Nutzung von verbotenen Adressen steht nicht im Gegensatz zum Ansatz der Nutzung einer geschlossenen Strategie, die ja vorsieht das alles, was nicht ausdrücklich erlaubt ist, verboten ist. Die Möglichkeit verbotene Adressen anzugeben wird nur genutzt, um eine die Menge der erlaubten Adressen effizienter beschreiben zu können. Wenn ein Nutzer z.B. alle Server mit den Adressen 141.12.133.* mit Ausnahme der Adresse 141.12.133.166 nutzen darf, dann kann diese Menge mit Angabe der verbotenen Adresse leicht beschrieben werden. Ohne die Möglichkeit die verbotene Adresse anzugeben, müsste eine sehr lange Liste von erlaubten Adressen angegeben werden.

Es ist auch von Vorteil neben der expliziten Beschreibung der einzelnen Adressen, Aggregationen zuzulassen, wie *„all“* für erlaubte Adressen und *„none“* für verbotene Adressen, wenn nicht die Reduzierung für eine gegebene Menge von Adressen notwendig ist:

```
PermDestinations ::= SEQUENCE  {
     allowedDests     Destinations
     forbiddenDests   Destinations
}

Destinations ::= SET OF Destination

Destination ::= OCTET STRING
```

Die Definition von Erweiterungen so wie sie hier vorgestellt ist, dient als Beispiel für Elemente und Nutzungsbedingungen die von im Kontext des Roaming von Interesse sein können. In der hier von mir vorgeschlagenen Struktur, habe ich allgemeine Nutzungsbedingungen definiert (z.B., quality of service), die für alle Dienste nützlich sind, und Dienst-spezifische Nutzungsbedingungen (z.B., Tage). Es ist auch möglich die Nutzungsbedingungen zu wechseln von allgemein zu Dienst-spezifisch und umgekehrt. Des Weiteren ist es ebenso möglich, diese Nutzungsbedingungen für beide Stufen zu

definieren. Bei einem solchen Ansatz wäre wieder eine “most specific wins” Strategie anwendbar. Eine QoS 100 kbps Bedingung für den MIP–Dienst würde eine allgemeine Bandbegrenzung auf 500 kbps als Nutzungsbedingung überschreiben.

Das Konzept der Definition von Möglichkeiten Zertifikatserweiterungen als Zugriffsrechte der Nutzer für mobile Kommunikationsdienste kann eingesetzt werden um die Menge der Zertifikate zu reduzieren. Eine solche Reduzierung kann durch die Einführung von Zugriffsrechte-Hierarchien erreicht werden, was beispielsweise bedeutet, dass ein bestimmtes Recht verschiedene andere Rechte impliziert. Dies ist Möglich durch die Einführung von teilweise geordneten (Halbordnungen) Mengen.

Solch eine Reduzierung kann durch die Einführung von Erlaubnis- bzw. Genehmigungshierarchien erreicht werden. Das würde bedeuten, dass eine Genehmigung automatisch bestimmte andere Genehmigungen impliziert. Dies ist möglich, in dem man Genehmigungen als halbgeordenete Mengen durch Einführung von Senior-Junior Beziehungen zwischen den entsprechenden Genehmigungen anordnet.

Wenn ein Nutzer eine mächtige Genehmigung erhält, erhält er damit auch alle Genehmigungen, welche diese impliziert.

Wenn man eine Beziehung voraussetzt, in welcher die Genehmigung für den Dienst MIP die Genehmigung für den Dienst MIP VPN impliziert, dann erhält eine Nutzer mit der Genehmigung für den Dienst MIP automatisch auch den MIP VPN Dienst ohne das dies explizit in seinem Nutzerzertifikat aufgeführt ist. Das Konzept von Senior-Genehmigungen, die automatisch Junior-Genehmigungen setzt voraus, das die Nutzungsbedingungen, die mit den Junior-Genehmigungen verbunden sind mit denen der Senior-Bedingungen übereinstimmt. Wenn die Nutzungsbedingungen der Junior-Genehmigungen von denen der Senior-Genehmigungen abweichen, müssen sie explizit definiert werden. Diese explizite Definition erlaubt es die aus der Senior-Genehmigung resultierende Nutzungsbedingung der Strategie „most specific wins“ folgend zu überschreiben.

Die in diesem Abschnitt präsentierte Lösung resultiert aus der Anforderung einer hohen Geschwindigkeit bei der Autorisierung. Wenn ein kontaktierter ISP die Identität des Nutzers überprüft, kann er sofort der Möglichkeiten des Nutzers gewahr werden. Dieses Wissen, kann dann für die Autorisierungsentscheidung und deren Durchsetzung eingesetzt werden, ohne dass es nötig ist im Rahmen der Autorisierung zusätzlich mit einer dritten Partei zusammenzuarbeiten. Dies gilt unter der Voraussetzung, dass auch aus anderen Gründen keine dritte Partei involviert werden muss. Der kontaktierte ISP Wird die im Nutzerzertifikat enthaltenen Genehmigungen nur erteilen, wenn er das Zertifikat auch verifizieren kann. Wenn der kontaktierte ISP keinen Validierungspfad vom Nutzerzertifikat zu einem seiner Vertrauensanker herstellen kann oder nicht in der Lage ist, die zur Verifikation des Nutzerzertifikates benötigten Algorithmen auszuführen, dann ist die sofortige Erteilung der in den Zertifikatserweiterungen enthaltenen Genehmigungen nicht möglich. In diesem Fall delegiert der kontaktierte ISP die

Authentifizierung an den RSP. Prinzipiell kann der kontaktierte ISP zwar in der Lage sein, das Nutzerzertifikat mit den im Zertifikat enthaltenen Genehmigungen zu lesen, eine sofortige Erteilung der Genehmigungen ohne Verifikation des Zertifikates stellt aber ein Risiko da, weil es sich zunächst um nicht vertrauenswürdige möglicherweise gefälschte Angaben handelt. Erst nachdem er ein positives Ergebnis der vom RSP durchgeführten Verifikation erhalten hat, sollte der ISP die Genehmigungen wie sie im Zertifikat stehen erteilen.

Da der in diesem Abschnitt gemachte Ansatz durch die Anforderung einer hohen Geschwindigkeit bei der Autorisierung entstanden ist, wurden die Erfüllung der in Abschnitt 7.1 aufgestellten Anforderungen bisher nicht betrachtet. Im Folgenden wird daher analysiert, in wieweit diese Anforderungen erfüllt sind:

- **Effizienz der Autorisierungsentscheidung und Durchsetzung:** Die vorgeschlagene Lösung erlaubt es dem kontaktierten ISP sofort die Autorisierungsentscheidung zu treffen, das die Anfrage des Nutzers nach bestimmten Ressourcen direkt mit den in seinem Zertifikat festgehaltenen Genehmigungen ohne Interaktion mit dritten Parteien abgeglichen wird. Es ist dabei nicht notwendig in einer Datenbank nach Autorisierungsdaten zu suchen und es sind keine komplexen Prozesse notwendig, um die Autorisierung des Nutzers in Abhängigkeit von seiner Identität zu erreichen. Des Weiteren werden die Autorisierungsentscheidungen von derselben Partei getroffen, die sie auch durchsetzt.
- **Aktualität der Autorisierungsregeln:** Die Aktualität der Autorisierungsregeln hängt hier von der Aktualität der Zertifikate ab. Wenn ein Nutzer neue Genehmigungen erhält, muss ein entsprechendes neues Zertifikat ausgestellt werden. Die Zeit, die ein Nutzer auf seine neuen Genehmigungen warten muss, hängt also von der Zeit ab, die für die Erzeugung und Verteilung des neuen Zertifikates benötigt wird. Der Nutzer kann aber immer noch sein altes Zertifikat verwenden bis er das neue erhalten hat, um damit die im Vergleich zu den neuen Rechten eingeschränkten alten Rechte wahrzunehmen. Wenn die Rechte eines Nutzers eingeschränkt werden, wird das alte Zertifikat zurückgerufen und es wird ein neues Zertifikat ausgestellt. Wenn das alte Zertifikat zurückgerufen wurde bevor der Nutzer sein neues Zertifikat erhalten hat, entsteht das Problem, dass der Nutzer in dieser Zeit von der Nutzung aller im Zertifikat genehmigten Dienste ausgeschlossen ist. Auch die Dienste, zu deren Wahrnehmung er berechtigt ist, werden ihm dann nicht geboten.
- **Unterstützung von Flexibilität bezüglich der Zugriffskontrollregeln:** Jedes Mal, wenn ein Unternehmen oder ein ISP die Rechte eines Nutzers verändern wollen, müssen sie für ihn ein neues Zertifikat ausstellen. Wenn die Rechte eingeschränkt werden sollen, muss das alte Zertifikat zusätzlich zurückgerufen werden. Diese Vorgänge brauchen Zeit und verursachen Kosten. Bei Änderung der Zugriffsrechte für größere Mengen von Nutzern in kürzerer Zeit, werden die CRLs zusätzlich noch unverhältnismäßig groß. Um Kosten zu vermeiden sollten Unternehmen und ISPs die häufige Änderung der Zugriffsrechte vermeiden. Dies

könnte jedoch im Gegensatz zu gegebenen Notwendigkeiten stehen. Daher wäre ein hierfür passender Ansatz, die Rollen, die Nutzer innehaben, in Zertifikatserweiterungen ihrer Zertifikate zu schreiben. Dadurch müssten Änderungen der Rechte, die eine Rolle innehat, nicht automatisch in Neuausstellung von Zertifikaten für alle Nutzer, die diese Rollen innehaben resultieren. Allerdings muss der ISP bei diesem Ansatz immer Wissen, was für Rechte mit welcher Rolle verbunden sind. Wenn man davon ausgeht, dass diese Recht-Rolle Zuordnung für jeden ISP spezifisch ist, dann müssen zwischen allen ISPs die Zuordnungsdaten ausgetauscht werden, was wieder einen hohen Aufwand darstellt. Dieses Problem ließe sich Lösen, indem Zuordnungen von Genehmigungen bzw. Rechten zu Rollen als Standard festgelegt wird, an den sich alle ISPs halten. Dies stünde aber dann im Gegensatz zur Flexibilität bezüglich der Zugriffskontrollregeln.

- **Privatsphäre und Geheimhaltung:** Wenn die Rechte eines Nutzers in seinem Zertifikat nachzulesen sind, kennen alle Parteien, welche sein Zertifikat lesen können, alle seine Rechte. Dies kann zu Problemen im Rahmen der Wahrung der Privatsphäre führen, da mehr Information preisgegeben wird als nötig. Die komplette Autorisierungsinformation auf einen Nutzer bezogen wird immer an jede Partei gegeben, die das Zertifikat des Nutzers erhält, z.B. auch wenn es sich um eine zur Authentifizierung in einem Kontext außerhalb der Nutzung mobiler Dienste handelt. Immer alle Rechte eines Nutzers offen zu legen wird vermutlich nicht mit der Strategie von vielen Unternehmen oder ISPs vereinbar sein, da Informationen über die Zugriffsrechte von Mitarbeiten eines Unternehmens oder Kunden einen ISPs normalerweise dritten Parteien, die diese nicht benötigen auch nicht zugänglich gemacht werden. Der Schutz dieser Informationen kann bis zu einem gewissen Grad durch Pseudonym-Zertifikate, welche anstatt der wahren Identität eines Nutzers ein Pseudonym wie z.B. eine ID beinhalten, erreicht werden.
- **Protokoll:** Die Autorisierung ist verhältnismäßig einfach, da die Zugriffsrechte eines Nutzers in seinem Zertifikat stehen und mit diesem übertragen werden. Dies impliziert allerdings, dass die Menge der Rechte, die im Zertifikat steht, möglichst klein sein sollte, da das Zertifikat jedes Mal, wenn der Nutzer sich irgendwo authentisiert, übertragen wird. Es ist fraglich, ob komplexe feingranulare Autorisierungsregeln in Zertifikaten übertragen werden sollten. Wenn die Regeln vom Umfang her den Beispielen dieses Abschnitts entsprechen, dann ist ein zertifikatebasierter Ansatz machbar. Wenn ein Unternehmen oder ein ISP aber eine sehr viel voluminösere Beschreibung seiner komplexen Autorisierungsregeln benötigt, ist der Zertifikatebasierte weniger geeignet.
- **Sicherheit bei der Übertragung der Autorisierungsinformation:** Die Autorisierungsinformation ist wenn sie in Zertifikate eingebunden ist genauso geschützt, wie alle anderen Informationen in den Zertifikaten. Sie ist signiert von der Aussteller-CA. Die Daten können dementsprechend zwar gelesen und modifiziert werden, aber eine Modifikation würde bei Überprüfung der Signatur sofort entdeckt werden.

- **Verfall und Sperre von Autorisierungsinformation:** Zertifikate verlieren zu einem bestimmten vordefinierten Zeitpunkt ihre Gültigkeit oder werden zurückgerufen, woraufhin sie nicht mehr anerkannt werden. Wenn Autorisierungsinformation in Nutzzertifikaten enthalten ist, müssen sie jedes Mal, wenn sich die Rechte des Nutzers ändern zurückgerufen werden. Daher sollte die vorgegebene Gültigkeitsdauer dem Zeitraum angepasst werden, in dem die angegebenen Rechte wahrscheinlich gültig sind, um häufiges Sperren und Neuausstellen von Zertifikaten zu vermeiden. Dieser Zeitraum kann kürzer sein, als es bei Zertifikaten, die ausschließlich zur Authentifizierung benutzt werden, der Fall ist.

7.4 Konklusion

In diesem Kapitel wurde eine für das Szenario eines umherreisenden Nutzers, der z. B. als Angestellter eines Unternehmens auf Unternehmens-Ressourcen oder als Kunde eines ISPs auf dessen Ressourcen zugreifen will, eine Autorisierungslösung entwickelt, die den Geschäftsmodellen aus Kapitel 3 gerecht wird.
Dabei wurden zunächst die Anforderungen, welche die Autorisierungslösung erfüllen soll, aufgestellt. Danach wurde die Architektur der auf SAML basierenden, hier entwickelten Lösung dargestellt und auf eine besonders schnelle Autorisierung mittels einer entsprechenden Lösung eingegangen.

Da - wie in der Einleitung erwähnt - im Roaming Kontext die zusätzliche spezielle Anforderung eines möglichen SSO besteht, da ohne ein solches SSO eine unterbrechungsfreie Dienstenutzung nicht möglich ist, wird im nächsten Kapitel eine entsprechende Lösung entwickelt.

8 Single Sign On (SSO)

Obwohl die dritte Generation der Mobilfunkanbieter sich noch nicht im Markt durchgesetzt hat, d.h. von der breiten Masse genutzt wird, ist die Entwicklung der Systeme der vierten Generation bereits auf dem Weg, mindestens wenn man Forschung und Entwicklung betrachtet [EITO03]. Nach [MorMurMoc00, BemTeuPlaPeePed02] kann man voraussetzten, dass der Wert der Kommunikationssysteme in der Zukunft anders bemessen wird. Der Wert der heutigen Netze basiert hauptsächlich auf ihrem Datenübertragungskapazitäten, während der wesentliche Gewinn bei den Netzen der nächsten Generation von den Diensten, die sie anbieten, abhängt.

Aus der Sicht der Kunden bieten diese Dienste die Gelegenheit, verschiedene Anwendungen mit einem potentiell hohen Wert zu nutzen, indem sie Benutzerfreundlichkeit oder das Sicherheitsniveau erhöhen. Solche Anwendungen können Dienste eines einzelnen Anbieters oder die Kombination von Diensten von verschiedenen Anbietern darstellen. Des Weiteren können Nutzer wenn sie „online“ sind im Nachhinein Anwendungen benötigen, welche auf Dienste zurückgreifen, die von verschiedenen Dienstanbietern verwendet werden. Bei derartigen Interaktionen werden Dienstanbieter die Möglichkeit haben wollen, die Identität derjenigen Kunden, welche die Dienste in Anspruch nehmen wollen, zu überprüfen, z.B. um korrekt Abrechnungen erstellen zu können. Für den Endverbraucher ist es jedoch inakzeptabel, wenn er jedes Mal eine Interaktion durchführen muss, wenn ein Dienstanbieter seine Identität überprüfen will. Daher sind Systeme für die Unterstützung des Single Sign On (SSO) erforderlich [BemTeuHoe03].

Im Allgemeinen gibt es verschiedene Vorschläge für die Realisierung eines SSO. Jedoch im Kontext dieser Arbeit gibt es spezielle zusätzliche Anforderungen für ein SSO. Da mobile Endgeräte beschränkte Rechenkapazitäten und Funkverbindungen mit eingeschränkter Bandbreite haben, soll das mobile Endgerät des Dienstnutzers so wenig wie möglich in das Authentifikationsprotokoll involviert sein[26] Daher wird vorgeschlagen, dass ein Nutzer mittels Belegen authentifiziert wird, welche innerhalb des Protokolls generiert werden, in das der mobile Nutzer involviert ist. Danach aber wird dieser Beleg zu den korrespondierenden Dienstanbietern geschickt, um sie mit nötigen Daten zu versorgen, damit ein überprüfbarer Identitätsbeweis bereitgestellt werden kann. Solch ein Beleg oder Identitätsbeweis besteht im Wesentlichen aus einer authentifizierten Nachricht, welche die Überprüfung ihrer Herkunft erlaubt, d.h. man kann verifizieren, dass der Beleg von einer bestimmten Partei ausgestellt wurde, wie z.B. bei einer digital signierten Nachricht des Nutzers. Wenn eine solche Nachricht jedoch zur

[26] Es gibt Lösungen, bei denen Komponenten von SSO Systemen lokal auf der Maschine des Nutzers installiert sind [PasMit03]. In diesen Systemen führt der Nutzer eine erste Authentifikation mit der lokalen SSO-Komponente durch. Danach führt die SSO-Komponente auf dem lokalen System das Authentifikationsprotokoll mit dem SPI im Namen des Nutzers aus. Dieser Ansatz reduziert die Anzahl Interaktionen des Nutzers aber keinesfalls die Anzahl der Interaktionen der Maschine des Nutzers.

Authentifikation oder Identifikation generiert wird, dann muss darauf geachtet werden, dass der Beleg wirklich vom mobilen Nutzer für diesen Zweck erzeugt wurde und nicht Bestandteil einer „replay attack“[27] ist. Das bedeutet, dass der Beleg auf eine Art und Weise erzeugt werden muss, die zumindest mit einer hohen Wahrscheinlichkeit Angriffe erkennen lässt.

In dieser Arbeit wird angenommen, dass der mobile Nutzer seine Identität dadurch nachweist, dass er eine zertifikatebasierte Authentifikation einmal mit einem Dienstanbieter durchführt, z.B. mit einem ISP oder einem WLAN Provider (WLANP). In diesem Zusammenhang wird der Nutzen des TLS Protokolls bzw. des EAP-TLS gesehen [DieAll99, AboSim99]. Fordert der mobile Nutzer einen Dienst an, der von einem anderen ISP geleistet werden muss, dann wird eine überprüfbare Authentifikation erzeugt und im anfänglichen Authentifikationsprotokoll (TLS) gegen die ursprüngliche ausgetauscht. Diese Authentifikationsinformation wird vom Authentifikator an den entsprechenden ISP gesendet, der die Korrektheit erforderlichenfalls verifizieren kann. Hinsichtlich des Austauschs dieser Authentifikationsinformation wird vorgeschlagen, Authentifikationsversicherungen einzusetzen, wie sie im SAML Standard definiert sind.

Das oben ausgeführte Geschäftsmodell kennt unterschiedliche Parteien, die in das SSO involviert sind. Im Folgenden wird nun die Lösung für dieses Modell erklärt. Sie ist so gestaltet, dass sie den Anforderungen für Hochgeschwindigkeitsauthentifizierung genügt. Sie basiert weiterhin vollständig auf existierenden Standards, was einen verhältnismäßig niedrigen Aufwand und niedrige Kosten für eine Implementierung garantiert.

Hier wird nun kurz skizziert, wie die Situation im Dienstanbietermarkt sich erwartungsgemäß verändert. Dies soll der Tatsache Rechung tragen, das die Lösung, welche sich auf das in dieser Arbeit vorgestellte Geschäftsmodell bezieht, generisch genug ist, um auch für andere Modelle gültig zu sein. Eine wichtige Eigenschaft der SSO Lösung ist, dass sie auch auf andere Modelle angewendet werden kann. Weiterhin werden hier verschiedene Ansätze für SSO vorgestellt und die Gründe für die gewählte Lösung dargelegt. Weiterhin wird erklärt, wie SSO in das hier entwickelte Geschäftsmodell passt. und wie passende Authentifikations-„assertions“ für SSO auf der Basis von SAML zu erzeugen sind. Danach wird die Lösung hinsichtlich ihrer Eigenschaften bewertet, und es werden die Implementierungs-“impacts“ diskutiert.

8.1 Dienstebereitstellung in Mobilkommunikationsnetzen der Zukunft

Mit der Bereitstellung von zukünftigen Telekommunikationsnetzen werden sich die Rolle und die Wichtigkeit der dem Endkunden angebotenen Dienste verändern. Während in der Vergangenheit der Wert der Netze von ihrer Fähigkeit, Kommunikation zu gewährleisten,

[27] Für eine detailliertere Beschreibung der Unterschiede zwischen der Nachrichten-Authentifikation und der Entity-Authentifikation wird an [MenOorVan96] verwiesen In dieser Arbeit werden E*ntity Authentifikation* and *Identification* als Synonyme benutzt.

bzw. der Datenübertragungskapazität abhing, gehen die Netzbetreiber davon aus, dass sie in der Zukunft einen höheren Gewinn mit Diensten erzielen werden. Auch vom Endverbraucher wird erwartet, dass er in Zukunft spezifische Dienste, die für ihn nützlich sind und einen zusätzlichen Wert schaffen, in Anspruch nimmt. Die Geschäftsmodelle der Zukunft werden auf diesen Annahmen aufgebaut.

Nach heutiger Vorstellung gibt es verschiedene Dienste, welche in der Zukunft eine Rolle spielen könnten. Welche davon tatsächlich angenommen werden, muss sich herauskristallisieren. Diese Dienste beziehen sich beispielsweise auf folgende Bereiche:

- Unterstützung von Mobilität,
- Inhaltsabhängigkeitsunterstützung,
- Orts-spezifische Informationen,
- Value-added Information,
- Bereitstellung von Multi-Media Inhalt in Verbindung mit Technologie des „digital rights managements“,
- On-demand Bereitstellung von benötigter Information,
- On-demand Bereitstellung von benötigten Ressourcen, wie z.B. Speicherplatz oder Bandbreite,
- Sicherheitsbezogene Dienste.

Dienste sind das letzte fehlende Glied in einer integrierten Wertschöpfungskette, welche neue Geschäftsmodelle einleiten können. Nach der Erforschung und der Entwicklung von grundlegenden Netztechnologien, Plattformen und deren Verfügbarkeit als Produkt gehen Forschung und Entwicklung in die Richtung der Dienste.

Es besteht die Anforderung, dass Dienste so weit wie möglich unabhängig von der Technologie des darunterliegenden Netzes sind. Dies erlaubt, dass sie unabhängig von der zugrunde liegenden Netztechnologie auch in Szenarien, bei denen es um den übergangslosen Wechsel zwischen Netzen geht, Verwendung finden können. Auf der anderen Seite haben die zugrunde liegenden Netztechnologien doch Einfluss hinsichtlich “quality of service“, wie z.B. Bandbreite des verwendeten Zugangsnetzes. Diese Anforderungen scheinen gegensätzlich zu sein. Es gibt jedoch die klare Anforderung, dass Nutzer nicht in irgendeiner Weise von irgendwelchen technischen Eigenschaften der zugrunde liegenden Netztechnologie gestört werden sollen. Um dem Rechnung zu tragen, werden zukünftige Entwicklungen von einem Ansatz ausgehen, bei dem im Zentrum die Dienste stehen. Diese Vision geht einher mit neuen Einflüssen auf den Authentifikationsvorgang und Anforderungen an Dienstanbieter und deren Endkunden. Die Existenz von kombinierten Diensten, die über verschiedene Zugangsnetztechnologien verfügbar sind, benötigt eine vereinheitlichte Authentifikationsmethode und eine vereinigte und effiziente Lösung für SSO. Das bedeutet, dass die Authentifikation und das SSO transparent und abstrahiert vom zugrunde liegenden Netz sein sollten.

In zukünftigen Dienstumgebungen werden wir verschiedene Rollen haben, die in die Bereitstellung von Diensten involviert sind. Zuerst haben wir die Rolle eines einfachen

Diensteanbieters, der seine speziellen Dienste wie ein Produkt anbietet. Zweitens haben wir die Rolle eines Dienstesammlers, der wie ein Mittelsmann oder Vermittler zwischen den Endkunden und verschiedenen Diensteanbietern ist. Tatsächlich ist diese Aggregation von Diensten selbst auch ein Dienst. Neben der Aggregation können Dienstesammler auch andere grundlegende Dienste anbieten. Zusätzlich können Diensteanbieter persönlicher Nutzerpräferenzen zu anderen Diensten hinzufügen und sie so an die speziellen Nutzeranforderungen anpassen. Des Weiteren können sie im Auftrag des Endnutzers Dienste anderer Anbieter anfragen. Endnutzer können entweder eine direkte Beziehung mit einem Diensteanbieter haben oder sie können eine indirekte Beziehung über einen Dienste „Aggregator“ haben. Alle Rollen werden in Abbildung 151 dargestellt.

Da Diensteaggregatoren als Mittler zwischen Endnutzern und anderen Diensteanbietern fungieren, sind sie auch für die Rolle von Authentifikatoren geeignet. Das bedeutet, dass sie die Authentifikationsprotokolle ausführen und andere Diensteanbieter mit den entsprechenden Zusicherungen “assertions” außerhalb des Authentifikationsprotokolles, welches mit der zu authentifiziernden Einheit, dem Endnutzer, durchgeführt wird, versorgen. Der Diensteaggregator schickt anderen Diensteanbietern Zusicherungen für die Identität eines Endnutzers, um eine Lösung für das SSO bereitzustellen. Um es anders zu sagen, bei einem solchen Ansatz wird die anfängliche Entitätsauthentifikation von einem solchen Zwischenanbieter ausgeführt werden.

Für den Endnutzer sollten komplexe Beziehungen und technologische Bedingungen des Diensteanbieters und Dienste Aggregators transparent sein. Es darf nicht nötig sein, dass der Endnutzer sich darum kümmert. In manchen Fällen kann es erwünscht sein, dass der Endnutzer nicht einmal bemerkt, dass er Dienste von verschiedenen Anbietern nutzt. Dies bedeutet, dass Diensteanbieter oder Aggregatoren eine einheitliche Technologie für die Authentifikation und das SSO benutzen, um die Bedienerfreundlichkeit und Bequemlichkeit für den Endnutzer zu unterstützen.

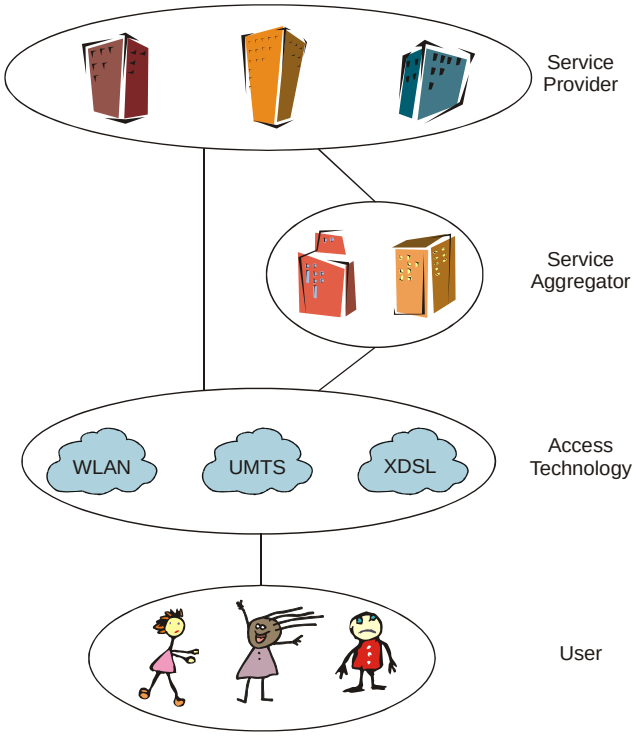


Abbildung 151: Dienst-Anbieter, Diensteaggregatoren und heterogene Zugangsnetze

8.2 Single Sign-On Lösung

Im Allgemeinen sind SSO-Lösungen benutzerfreundlich. In diesem Zusammenhang kann die Benutzerfreundlichkeit verschiedene Gesichter haben, wie z. B. reduzierte Benutzer-Interaktion oder schnellen Zugang zu Diensten, da automatisierte SSO Mechanismen schneller sind, als zusätzliche Authentifizierungen, die mit einem Nutzer durchgeführt werden müssen. Minimierung der Nutzerauthentifizierung durch höchstmögliche Reduzierung der Endnutzerinteraktionen, schneller Zugang zu Diensten, da automatisierte SSO Mechanismen schnelleren Zugang zu Diensten haben als zusätzliche vom Endnutzer ausgeführte Authentifikationsmechanismen. Es ist nicht nötig, für den Endnutzer viele verschiedene Passwörter und Logins zu verwalten. Praktisch bedeutet das “Single” bei SSO Systemen, dass der Endnutzer sich selbst oder das Endgerät nur ein einziges Mal authentifizieren muss. In dem hier vorgestellten Ansatz wird einen Schritt weiter gegangen mit der Forderung, dass auf den Endgeräten des Nutzers die notwendigen Geheimnisse, wie z.B. ein Passwort oder ein geheimer Schlüssel nur ein einziges Mal innerhalb des Authentifikationsprozesses[28] angewandt werden müssen. Diese Anforderung ist viel anspruchsvoller als die üblicherweise an SSO Systeme gestellten

[28] Nutzer oder Endgeräte können in den Authentifizierungsprozess durch den Austausch von Tokens —tickets or Authentifizierungszusicherungen — involviert sein, die von Diensteanbietern zur Autehntifizierung eingesetzt werden können. Diese Token können entweder die vollständige für die Authentifizierung benutzte Information enthalten oder einen Verweis zu einer Anfrage nach Authentifizierungsinformationen von einer dritten Partei.

Anforderungen, bei denen nur die Anzahl der Nutzerinteraktionen minimiert wird. Diese zusätzliche Anforderung trägt der Tatsache Rechnung, dass mobile Endgeräte begrenzte Rechenkapazitäten und Kommunikationskanäle für mobile Endgeräte eine begrenzte Bandbreite haben. Es wird hier weiterhin von der grundlegenden Anforderung ausgegangen, dass die Authentifikation der Entitäten schnell verlaufen muss. Das ist wichtig, da die SSO Lösung die Authentifikation Idealerweise schnellstmöglich gewährleisten soll.

In der Praxis gibt es viele verschiedene mögliche Lösungen für SSO Systeme [PasMit03, Clercq02]. Man unterscheidet drei wesentliche Grundmodelle für SSO Systeme:

- Das Pull - Modell,
- das Push - Modell und
- das Proxy - Modell.

Diese Architekturen werden in Abbildung 152, Abbildung 153 bzw. Abbildung 154 dargestellt. Die Modelle unterscheiden sich hinsichtlich der Anzahl der benötigten Interaktionen der unterschiedlichen Parteien und beeinflussen daher die für den Authentifikationsvorgang benötigte Zeit. Beim Pull Modell überprüft ein Authentifikator die Identität des Nutzers und erhält ein Ticket. Wenn der Nutzer später mit einem Dienstanbieter verbunden wird sendet er das Ticket zu dem entsprechenden Dienstanbieter. Auf Basis dieses Tickets fragt der Dienstanbieter beim Authentifikator nach einer Bestätigung der Identität des Nutzers. In einem Push Ansatz, erhält der Nutzer eine „authentication assertion“ vom Authentifikator. Diese „assertion“ wird dann zusammen mit der entsprechenden Nutzeranforderung zum Dienstanbieter gesandt. Das wohl bekannteste Beispiel für ein Push Modell bietet Kerberos [KohNeu93].[29] Beim Proxy Modell handelt der Authentifikator im Auftrag des Nutzers, indem er eine „authentication assertions“ zum Dienstanbieter schickt. Die hier präsentierte Lösung legt das Proxy-Modell zugrunde.

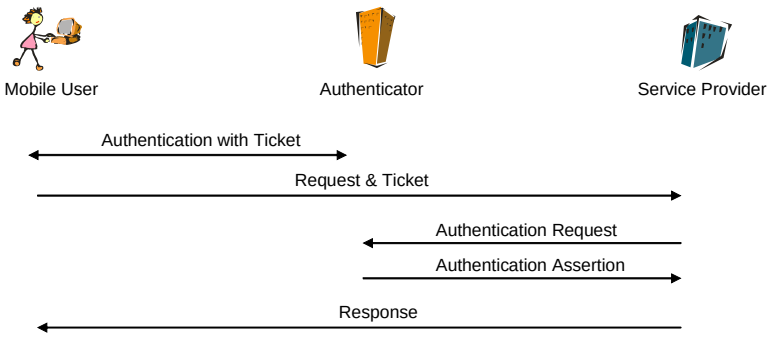


Abbildung 152: SSO Pull-Modell

[29] Die Authentifizierungszusicherungen die im Kerberos System verwendet werden, sind für gewöhnlich als Tickets bezeichnet [KohNeu93]. Diese Tickets sollten nicht mit den Tickets, die im pull Modell verwendet werden, verwechselt werden.

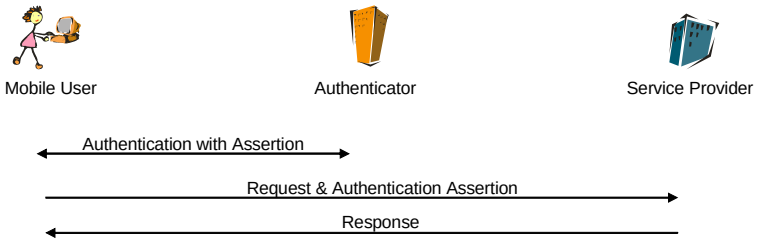


Abbildung 153: SSO Push-Modell

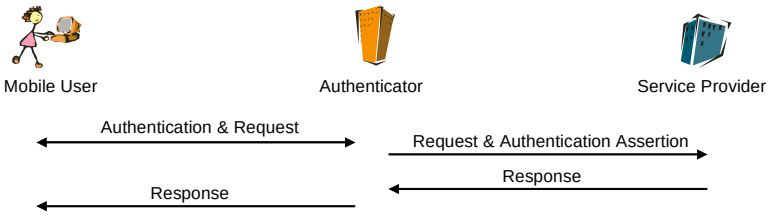


Abbildung 154: SSO Proxy-Modell

Des Weiteren können wir unterscheiden, ob die Identitäten der Nutzer, die bei einem SSO System behandelt werden, für einen Dienstanbieter spezifisch sind oder nicht. Im Allgemeinen kann ein Nutzer verschiedene Identitäten jeweils für jeden Dienstanbieter nutzen. Solch ein Ansatz wird für die Authentifikation von Entitäten benutzt, die auf Logins und Passwörtern basiert. Für solche Systeme sollten die Passwörter der Nutzer für die unterschiedlichen Dienstanbieter natürlich auch unterschiedlich sein. Wenn wir es mit zertifikatebasierter Authentifikation zu tun haben, müssen die Credentials nicht aus Gründen der Sicherheit unterschiedlich sein. Möglicherweise unterscheiden sie sich aus Gründen der Vertraulichkeit bzw. Wahrung der Privatsphäre, um dritten Parteien nicht zu ermöglichen, bestimmte Aktionen mit bestimmten Identitäten zu verbinden. Da das Ziel ist, die Zeit zu reduzieren, welche für die Authentifikation benötigt wird, wird hier vorausgesetzt, dass ein Nutzer nur ein Zertifikat für eine Instanz zur selben Zeit verwendet. Dies bedeutet, dass ein Nutzer nur ein Zertifikat besitzen muss, welches dann im Rahmen des hier verwendeten Geschäftsmodells überprüft wird, was positive Auswirkungen auf den benötigten Aufwand und die zeitliche Verzögerung hat.

Der Besitz von mehreren Zertifikaten macht es erforderlich, dass das Endgerät des Nutzers eventuell mehr als einen geheimen Schlüssel verwaltet. Weiterhin sei hier angemerkt, dass die Nutzer andere Zertifikate verwenden können, wenn sie sich später in anderen Sitzungen erneut authentifizieren.

Ein anderer Aspekt der hier angesprochen wird, bezieht sich auf die Tatsache, dass der Nutzer entsprechende Dienste von einem Anbieter entweder in direktem Kontakt oder über eine dritte Partei wie einen Diensteaggregator oder Stellvertreter (Proxy) anfordert,

welcher im Auftrag des Nutzers die eigentliche Anforderung stellt. Die Art und Weise wie von wem die Anforderung gestellt wird, hängt von dem jeweiligen Dienst ab. Im Falle eines Nutzers, der die gewünschten Dienste selbst in direktem Kontakt mit dem Dienstanbieter anfordert, muss der Nutzer den Dienstanbieter mit entsprechender Authentifizierungsinformation wie z.B. einem Ticket oder einer Authentifizierungs „assertion“ versorgen, welche die Überprüfung seiner Identität erlaubt, wie z. B: in einer auf dem Push-Modell oder auf dem Pull-Modell basierenden Lösung.
Eine wichtige Anforderung an SSO Systeme ist, dass sie sicher sein sollen, d.h. dass es nicht möglich sein soll für Angreifer/Gegner eine falsche Identität anzunehmen bzw. vorzutäuschen und andere Nutzer zu maskieren. Dienstanbieter, die einen SSO Dienst anbieten, wollen auf die Korrektheit der Identität, die innerhalb des SSO Systems festgestellt wurde, vertrauen. Ganz allgemein darf ein SSO System auf unterschiedlichen Vertrauensmodellen hinsichtlich des Vertrauens des Diensteanbieters zum Authentifikator beruhen.

Uneingeschränktes Vertrauen:
In diesem Modell vertraut der Diensteanbieter der Nutzeridentität, wie sie vom Authentifikator festgestellt ist. In diesem Fall wird keine Verifikation der Nutzeridentität vom Diensteanbieter gefordert. Dieses Vertrauensmodell wird im Kerberos System zu Grunde gelegt.

Eingeschränktes Vertrauen:
Der Diensteanbieter unterstellt, dass die Nutzeridentität, wie sie von Authentifikator festgestellt ist, korrekt ist, wenn der Authentifikator ein Nutzer spezifisches Geheimnis liefern kann. Damit ist gemeint, dass der Diensteanbieter vom Authentifikator überzeugt ist, dass dieser diese Geheimnisse nicht missbraucht und nicht mit einem Gegner einen schädlichen maskierten Angriff im Geheimen abspricht. Dieser Grad des Vertrauens ist erforderlich, wenn ein Authentifikator Nutzerpasswörter, die für SSO Systeme genutzt werden, betreut.

Kein Vertrauen:
Der Diensteanbieter hat kein Vertrauen darein, dass der Authentifikator immer korrekte Authentifikationsversicherungen liefert. Daher sollten die vom Authentifikator generierten Authentifikationsversicherungen einige nicht abstreitbare Daten enthalten, die vom Nutzer erzeugt sind. Dies soll garantieren, dass der Authentifikator nur dann mit Erfolg Ansprüche an einen Nutzer stellen kann, wenn er diesen vorher wirklich authentifiziert hat und von ihm einen Beweis seiner Identität hat, der von anderen nachvollzogen werden kann. Dieses Modell wird in der vorliegenden Lösung zu Grunde gelegt.

Ein Modell, welches kein Vertrauen als Grundlage hat, ist den anderen vorzuziehen, wenn Authentifikator und Diensteanbieter einander nicht kennen und somit nicht angenommen werden kann, dass sie sich vertrauen. Außerdem gibt es keine Geheimnisse zwischen Nutzern, Authentifikatoren und Diensteanbietern. Dies ist aus Sicht des Nutzers in der Praxis von Vorteil. Weiter hat dies auch einen Vorteil aus der Sicht der Sicherheit. Da es

keine Geheimnisse gibt, sind Missbrauch der Geheimnisse und maskierte Angriffe weitgehend ausgeschlossen.

Wenn die Nutzeridentität dem Diensteanbieter in einer Authentifikationsversicherung entweder vom Nutzer im Push-Modell oder vom Authentifikator im Proxy- und Pull-Modell mitgeteilt ist, dann sind einige zusätzliche Schutzmaßnahmen der Authentifikationsversicherung notwendig. Diese sollten sich zumindest auf die Integrität der Authentifikationsversicherung konzentrieren.

Die hier vorgeschlagene Lösung basiert auf dem Proxy-Modell. Sie ist dargestellt in Abbildung 155. Diese Lösung kann immer angewendet werden, wenn der Dienstanbieter keinen direkten Kontakt zum Endnutzer benötigt. Immer wenn möglich ist eine auf dem Proxy-Modell basierende Lösung vorzuziehen, da die Anzahl der Interaktionen niedrig ist. Des Weiteren ist der Nutzer nicht in den Austausch von Authentifikationsversicherung involviert. Dieses Argument trägt der geringeren Übertragungs-Bandbreite der Verbindung des Nutzerendgerätes und der Hochgeschwindigkeitsanforderung Rechnung. Das Ziel ist es, den Beweis, der durch das vom Nutzer verwendete Endgerät erzeugt wird, bei der ersten Authentifikation des Nutzers, weiter zu nutzen. Dieser Beweis besteht im Wesentlichen aus einer digitalen Signatur, bei der der Nutzer „challenge“-artige Daten signiert. Sowohl Daten als auch Signatur werden im Rahmen des Authentifikationsprotokolls für SSO Zwecke erzeugt.

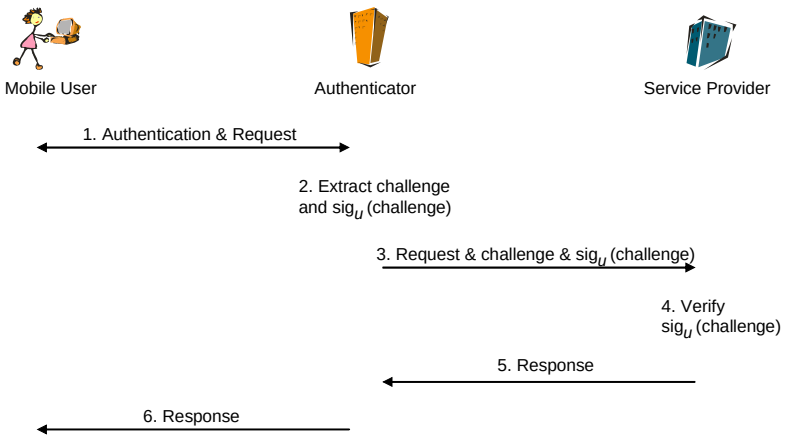


Abbildung 155: Authentifikation und SSO auf digitalen Signaturen basierend

Die Signatur kann dann als überprüfbare Authentifizierungsinformation benutzt werden und so in Szenarien angewandt werden, in denen keine Vertrauensbeziehung zwischen Dienstanbieter und Authentifizierer besteht. Der Authentifizierer generiert dann eine Authentifikationsversicherung, welche diesen Beweis nutzt. Des Nutzers anfängliche Authentifikation und die Authentifikationsversicherung basieren auf derselben Identität. Die Authentifikationsversicherung sind zusätzlich mit der Signatur des Authentifikators gesichert.

Damit Authentifikationsversicherung mit überprüfbaren Beweisen nicht von potentiellen Angreifern in Wiederholungsangriffen missbraucht werden können, müssen diese Beweise zusätzliche Daten enthalten, die eine Erkennung dieser Art von Attacken erlauben[30]. Die digitale Signatur sollte klar und in einer nicht fälschbaren Art und Weise von der Identität, die an der initialen Authentifikation beteiligt war, abhängen. Dies beugt der Wiederbenutzung solcher Beweise vor, d.h. Authentifizierer X kann die digitale Signatur, die ein Nutzer für Authentifizierer Y erzeugt hat nicht weiterverwenden.

Des Weiteren sollte ein Zeitwert, welcher den Zeitpunkt der initialen Authentifikation enthält, digital vom Nutzer signiert werden. Dies ermöglicht es den Authentifizierern, alte Signaturen des Nutzers zu erkennen. Der Zeitwert wirkt hier als Einmalwert (nonce), was hilft, wiederholtes Nutzen von alter Authentifizierungsinformation, welche Signaturen enthält, zu erkennen. Wenn der Dienstanbieter ein paar Kopien dieser Beweise in seiner lokalen Datenbank für einen angemessenen Zeitraum behält, können diese u. a. zum Erkennen von Angriffen verwendet werden. Es kann jedoch Fälle geben, in denen die Wiederholung von bereits verwendeten Beweisen innerhalb eines bestimmten Zeitraums erwünscht ist und daher erlaubt sein sollte.

In der hier vorgeschlagenen Lösung wird die Signatur als Beweis zur Nutzerauthentifikation genutzt. Die Signatur, welche als Beweis bei der Authentifikation des Nutzers eingesetzt wird, wird aus dem Handshake des TLS Protokolls extrahiert [DieAll99]. Diese Signatur wurde vom Nutzer erzeugt und kann von anderen Parteien überprüft werden.

Um mit genau einer Signatur, welche im Endgerät des Nutzers erzeugt wurde, auszukommen, spielt diejenige Partei, welche das TLS Protokoll fährt, die Rolle des Authentifizierers. In unserem Szenario wird diese Rolle vom ISP ausgefüllt. Diese Partei generiert dann eine Authentifikationsversicherung, welche zu anderen Dienstanbietern gesendet wird, so z.B. zum RSP zwecks Generierung von VPN Schlüsseln. Diese Authentifikationsversicherung wird unter Verwendung des SAML Standards implementiert.

8.3 Szenario “Sicheres Roaming für Endnutzer”

Die primäre Absicht dieses Abschnitts ist es klarzustellen, wie die hier vorgestellten Ergebnisse betreffend Authentifikation, Single Sign-On, Autorisation und PKI Ergebnisse zueinander in Beziehung stehen und wie sie in einer Implementierung des Geschäftsmodells technisch zusammenspielen.

In diesem Abschnitt wird ein Szenario beschrieben, welches auf dem in Kapitel 3.3 vorgestellten Geschäftsmodell beruht. Das Ziel des hier vorgestellten Modells ist es, sicheres Roaming als Dienst für Endnutzer zu Verfügung zu stellen. Dies erlaubt mobilen

[30] Externe Parteien und Authentifikatoren selbst sind potentielle Kandidaten für diese Kontrahenten.

Nutzern, Zugang zum Internet über verschiedene ISPs zu erlangen. Des Weiteren können mobile Nutzer dahingehen eingeschränkt werden, dass sie nur Verbindungen zu gegebenen Zielen, wie z.B. dem Unternehmen, in dem sie angestellt sind, aufbauen können.
Zusätzlich kann eine VPN Verbindung zwischen dem mobilen Nutzer und seinem Unternehmen vom RSP unterstützt werden. Solch eine sichere Verbindung kann sogar erzwungen werden, indem keine anders geartete Verbindung zum Unternehmen zugelassen wird. Hierbei können PKI Dienste auch von Komponenten bereitgestellt werden, die sich im Unternehmen befinden. Dieses Modell zeigt Abbildung 156.

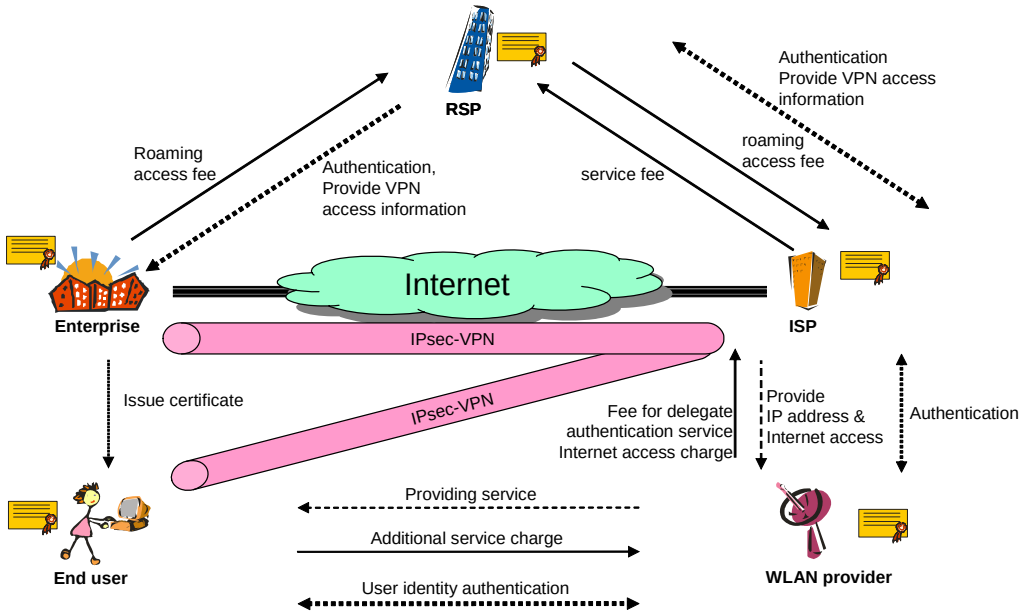


Abbildung 156: Überblick über das Modell Roaming mit VPN Zugang

In Abbildung 157 sind die involvierten Parteien und einige ihrer Interaktionen vereinfacht dargestellt. Bevor das Szenario beschrieben wird, erfolgt eine Vorstellung der involvierten Parteien. Es gibt den mobilen Nutzer, einen „access point“, einen kontaktierten ISP, zwei RSPs, und das Unternehmen, bei dem der mobile Nutzer arbeitet. Es sind zwei RSPs involviert, da nicht vorausgesetzt werden kann, dass der kontaktierte ISP und das Unternehmen Kunden desselben RSPs sind. Es wird also der allgemeinere Fall betrachtet, dass Unternehmen und ISP unterschiedlichen RSPs verbunden sind. Das Zugangsnetz und der kontaktierte ISP hingegen gehören zu derselben RSP Domäne, obwohl der mobile Nutzer und das Unternehmen einer anderen RSP Domäne angehören. Des Weiteren wird vorausgesetzt, dass die unterschiedlichen RSPs in irgendeiner Weise eine Übereinkunft hinsichtlich des Roamings ihrer Kunden getroffen haben, so dass die Kunden beider RSPs innerhalb und zwischen beiden Domänen „roamen“ können.

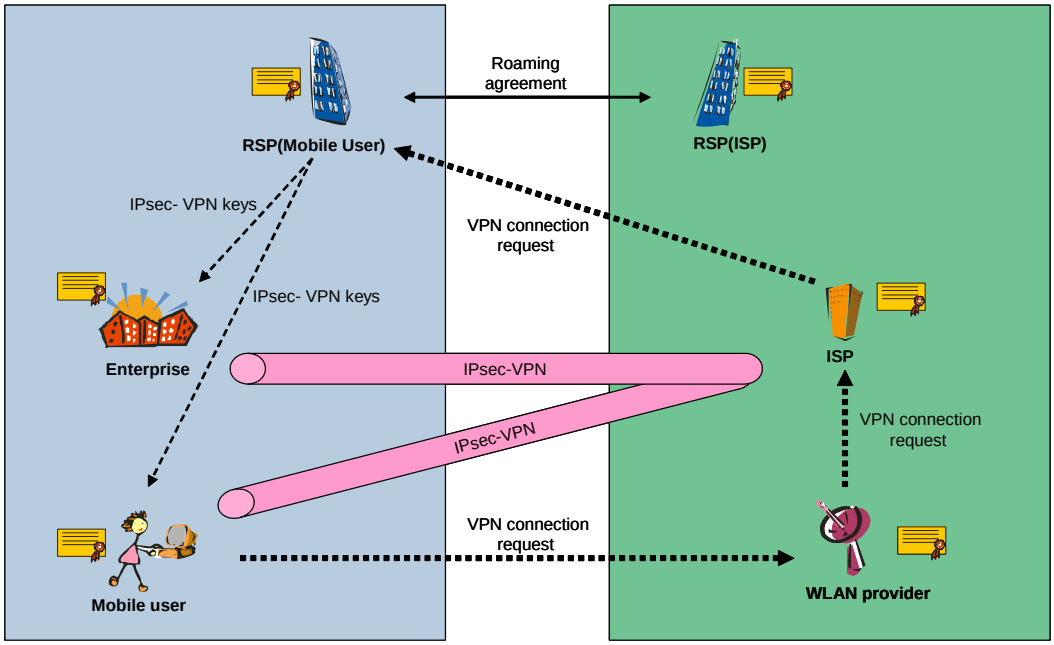


Abbildung 157: Parteien und Ihre Interaktionen

Es wird nun angenommen, dass die involvierten Parteien die Komponenten bereitstellen, die für die Lösung erforderlich sind. Wie in Abbildung 158 dargestellt, betreibt der kontaktierte ISP einen Authentifizierungsserver[31], der für beidseitige Authentifikation verwendet wird, und der RSP sowie das Unternehmen stellen spezifische PKI Server bereit, welche die Verifikation der Zertifikate durch Konstruktion der Zertifikatspfade bzw. deren Überprüfung unterstützen.

In Abschnitt 8.1 wurden abstrakte Rollen, so wie Dienstanbieter und Dienste-Sammler vorgestellt. Im Kontext des hier zugrunde gelegten Geschäftsmodells wird die Funktionalität des ISPs der abstrakten Rolle des Dienst-Sammlers zugeordnet, welcher in einem allgemeineren Sinn als Mittler zwischen Nutzer und Dienstanbieter fungiert. Im Geschäftsmodell entspricht der ISP einem Vermittler von Diensten für deine Kunden. Der ISP füllt die abstrakte Rolle eines Diensteaggregators nicht wegen seiner Kernfunktionalität – der Bereitstellung des Internetzugangs - aus, sondern weil er seine Kunden zusätzliche Dienste anbietet, wie z.B. Proxy-Unterstützung für Verteilung von VPN Schlüsseln. Des Weiteren spielt der RSP die Rolle eines Dienstanbieters, wie in Abschnitt 8.1 beschrieben. In dem hier vorgestellten Geschäftsmodell, führt der ISP die Authentifikation des Nutzers durch. Da der ISP verantwortlich ist, für die Authentifikation des Nutzers, ist er auch ein guter Kandidat für einen Authentifizierer in einem SSO System. In diesem Zusammenhang stellt der ISP als Authentifizierer Informationen für andere Dienstanbieter bereit, so wie den RSP.

[31] Der Authentifizierungsserver muss nicht notwendigerweise vom kontaktierten ISP betrieben werden. Er kann stattdessen auch vom WLAN Provider betrieben werden.

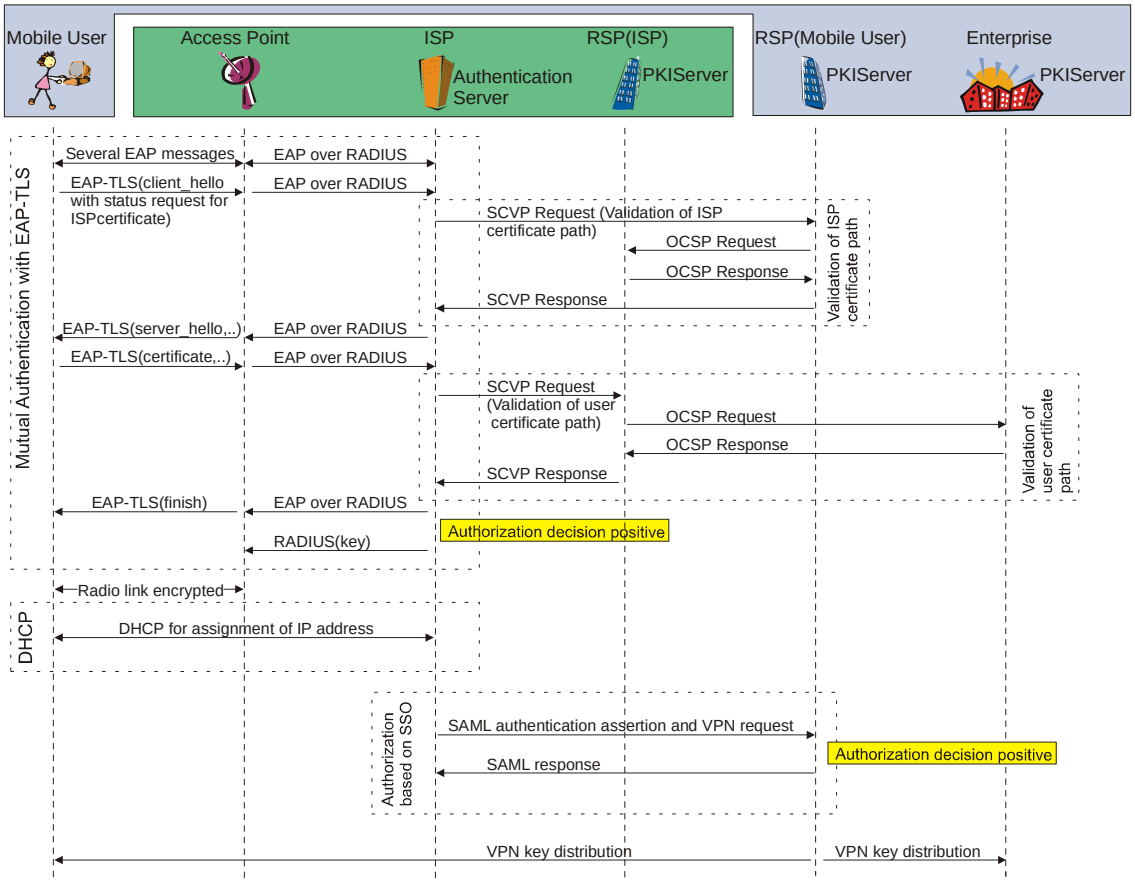


Abbildung 158: Vereinfachter Nachrichtenfluss im Modell Roaming mit VPN Zugang

Der ISP als Mittler rechtfertigt ebenso die Entscheidung für das SSO Proxy-Modell. Authentifikationsversicherungen werden generiert vom ISP. Diese müssen zwischen ISP und RSP ausgetauscht werden. Es kann vorausgesetzt werden, dass die Kanäle zwischen diesen Parteien über eine genügend hohe Bandbreite verfügen, während die Bandbreite der Kanäle zwischen dem Nutzer und dem ISP eher niedrig sein können. In der vorgeschlagenen Lösung werden die Authentifikationsversicherungen, die nur zwischen ISP und RSP ausgetauscht. Ein weiterer Vorteil eines SSO Proxy-Modells ist, das die Anzahl der Nutzer und Nutzergeräte Interaktionen reduziert werden. All diese Aspekte unterstützen die Erfüllung der Hochgeschwindigkeitsanforderung aus dem Geschäftsmodell an die Lösung.

Da ISP und RSP sich auf Grund ihrer wirtschaftlichen Interessen nicht unbedingt gegenseitig vertrauen, sind überprüfbare Authentifizierungsversicherungen, die vom Nutzer signierte Angaben beinhalten, nützlich. In einem Roaming Szenario kann man nicht voraussetzen, dass der Nutzer den kontaktierten ISP kennt, und daher kann man nicht fordern, dass der Nutzer diesem ISP hinsichtlich irgendwelchen Missbrauchs vertraut. Andererseits füllt der ISP hier die Rolle des Authentifizierers aus. In dem hier gemachten Vorschlag für SSO, erhält der ISP keinerlei geheime Information vom Nutzer.

Des Weiteren kann der ISP nur gültige Authentifikationsversicherungen erstellen, wenn er zuvor eine gültige digitale Signatur vom Nutzer erhalten hat.

Ein mobiler Nutzer versucht, eine VPN Verbindung zu seinem Unternehmen zu etablieren, über ein Zugangsnetz aus einer fremden RSP-Domäne wie in Abbildung 158 zu sehen ist. Der Nachrichtenfluss ist dabei sehr vereinfacht. Das Ziel dabei ist es, dass der RSP des Nutzers, dem vom Nutzer und vom Unternehmen getraut wird, VPN Schlüssel generiert und diese an den Nutzer und das Unternehmen weiterleitet.

Zu Beginn der Interaktion führen der Nutzer und der Authentifikationsserver des ISP eine beidseitige Authentifikation aus. In Abbildung 158 wird die Authentifikation durch EAP-TLS in Kombination mit EAP über RADIUS skizziert. Die benutzten Prinzipien zur beidseitigen Authentifikation sind weiter oben in Kapitel 4 und in Kapitel 2 bereits beschrieben. Wenn der mobile Nutzer den kontaktierten ISP authentifiziert, ist es erforderlich, den kompletten Zertifikatepfad zu verifizieren, der die Kette von Zertifikaten vom Zertifikat des ISP bis zum Zertifikat des Zertifikaterzeugers, dem der Nutzer traut, enthält. Der Einsatz von X.509 Zertifikaten [X.509] wird hierbei angenommen. Die Verifikation eines Zertifizierungspfades heißt, diesen Pfad zu konstruieren und die Korrektheit jedes Zertifikates hinsichtlich einer Sperre zu prüfen. Da der mobile Nutzer keine Internetverbindung hat, kann er selbst die Verifikation des Zertifikatepfades nicht durchführen. Als Abhilfe wird dies an einen PKI Server delegiert. Die Delegation dieser Aufgabe wird im TLS Handshake-Protokoll initiiert (Client-Hello). In Kapitel 6 beschrieben, wie dieses Problem mit PKI Servern gelöst werden kann, welche die Pfadvalidierungsarbeit für den Nutzer durch Verwendung von SCVP [MalHouFre03] und OCSP [MyeAnkMalGalAda99] ausführen. Da der mobile Nutzer seinem eigenen RSP vertraut, wird die Pfadvalidierungsarbeit an diese Partei, wie in Abbildung 158 zu sehen ist, delegiert. Die PKI Server können auch für die Identifikation in entgegengesetzter Richtung eingesetzt werden. In diesem Fall wird die Validierung zum RSP des ISPs delegiert. Diese ist wiederum in Abbildung 158 dargestellt. Das Ergebnis ist ebenfalls in Kapitel 6 beschrieben. Wenn die Authentifikationsresultate negativ sind, wird die Verbindung zwischen dem Nutzer und dem kontaktierten ISP oder “access point “ getrennt. Wenn die Authentifikation positiv ist, dann entscheidet der kontaktierte ISP, ob der mobile Nutzer die Erlaubnis bekommt, eine Internetverbindung zu bekommen oder nicht. Dies erfolgt dadurch, dass der kontaktierte ISP mit der notwendigen Autorisationsinformation im Nutzerzertifikat bereitgestellt wird. Einzelheiten zur Autorisierung entnehme man hierzu Kapitel 7.

Bei positiver Autorisierung wird die Funkverbindung zwischen dem Nutzer und dem “access point” verschlüsselt und der Nutzer erhält Zugang zum Internet mit einer IP Adresse durch das DHCP Protokoll. Wenn der kontaktierte ISP nun bemerkt, dass der Nutzer eine VPN-Verbindung zu seinem Unternehmen herstellen will, versorgt er den RSP des Nutzers mit Informationen hinsichtlich der Anforderung des Nutzers sowie Authentifikationsinformationen. Diese Authentifikationsinformation dient dem SSO: Der RSP nutzt sie, den Nutzer wieder ohne irgendeine Aktion von diesem zu identifizieren. In Kapitel 7 wird bereits vorgeschlagen den SAML Standard einzusetzen. In Abschnitt 8.4.2

wird dann gezeigt, wie eine SAML Nachricht erzeugt sein sollte, die eine Signatur einer „challenge“-artigen Nachricht, welche aus dem SSL-Handshake Protokoll extrahiert wurde, enthält. In Abbildung 158 sind die Nachrichten für die Autorisation und SSO aufgezeigt, die nach den DHCP Nachrichten auszutauschen sind. Diese Nachrichten können auch früher direkt nach dem Nachrichtenblock für die Validierung des Zertifizierungspfades des Nutzers ausgetauscht werden.

Abschließend prüft der RSP, ob der Nutzer dien Erlaubnis hat, den VPN Service des RSP anzufordern, in dem er die Erlaubnis im Zertifikat des Nutzers verifiziert. Wenn das Ergebnis Autorisationsentscheidung positive ist, generiert der RSP die VPN Schlüssel und sendet sie an den Nutzer und das Unternehmen. Der RSP kann die VPN Schlüssel sofort an den Nutzer senden, wie in Abbildung 157 gezeigt. Eine andere Möglichkeit wäre es, die VPN Schlüssel über den ISP zu versenden. Dies ist kein Umweg, da VPN Schlüssel im Netzwerk immer über den ISP laufen. Es muss jedoch für jeden Weg sichergestellt werden, dass der ISP sie erhält.

8.4 SSO basierend auf existierenden Standards

Um SSO auf der Basis von Zertifikaten zu erhalten, wird hier eine Lösung vorgeschlagen, welche die digitale Signatur ausnutzt, die für die Authentifikation im TLS Protokoll erzeugt wird. Es wird angenommen, dass die anfängliche Authentifikation zwischen Nutzer und ISP durch TLS ausgeführt wird. Die Authentifikationsinformation, die vom ISP erhalten wird und von einer anderen Partei verifiziert wird, wird an den RSP als eine SAML Nachricht weitergegeben. Die Kombination dieser zwei Mechanismen erlaubt es, SSO für einen mobilen Nutzer und seine Hardware zu realisieren.

Es wird nun zuerst beschrieben, wie die Authentifikationsinformation als Teil einer Handshake Nachricht durch TLS erzeugt wird. Danach wird erklärt, wie Authentifikationsversicherungen aufgrund der Authentifikationsinformation kreiert und vom kontaktierten ISP zu einem Dienstanbieter wie RSP unter SAML weitergeleitet werden.

8.4.1 TLS basierte Evidenz Generierung

Die Hauptziele von TLS sind es, die End-zu-End-Privatspähre und Datenintegrität zwischen zwei kommunizierenden Parteien durchzusetzen. Sie bietet jedoch auch die Möglichkeit der Entitätsauthentifikation, die in einem gewissen Sinne streng mit der Authentifikation von Datenursprüngen und Nachrichtenintegrität verknüpft ist. Bevor die auszutauschenden Daten durch das TLS „record“-Protokoll geschützt werden können, wird der Schutzmechanismus im TLS Handshake Protokoll übertragen. Die Entitätsauthentifikation zwischen einem Nutzer und einem Authentifikationsserver wird im TLS Handshake Protokoll ausgeführt. Die hier betrachtete Lösung SSO beruht auf digitalen Signaturen, die innerhalb des TLS Handshake Protokolls generiert und ausgetauscht werden. Der TLS Standard beschreibt diesen Handshake wie in Abbildung 159 [DieAll99]:

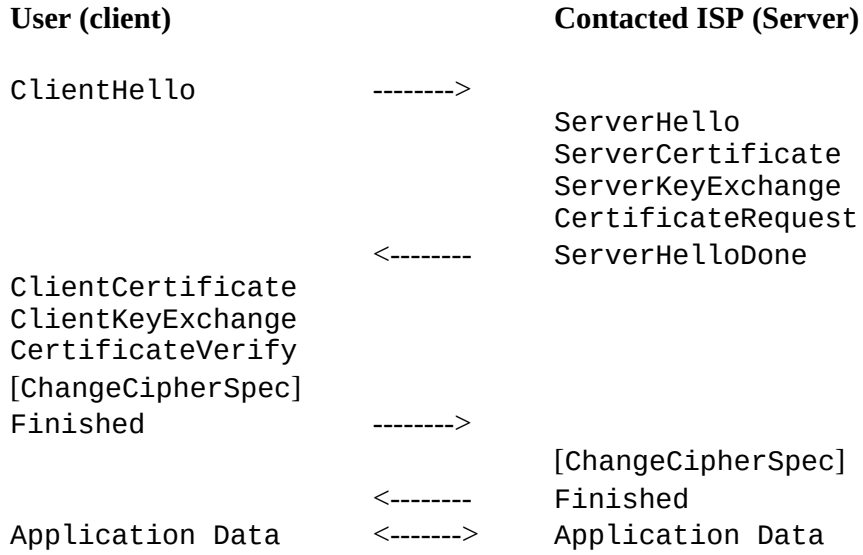


Abbildung 159: Das TLS Handschlag-Protokoll[DieAll99]

Im TLS Handshake Protokoll signiert der Nutzer eine Nachricht aus allen verketteten Handshake Nachrichten von der `ClientHello` Nachricht bis zur `CertificateVerify` Nachricht, wie in Abbildung 133 gezeigt, indem er seinen privaten Schlüssel benutzt. Diese Signatur wird dann in der `CertificateVerify` Nachricht ausgetauscht. Es wird angenommen, dass nur der Nutzer zu seinem privaten Schlüsseln zugreifen kann und sonst niemand, so dass niemand sonst die zugehörige Signatur generieren kann. Alle anderen Parteien können den Ursprung der Nachrichten mit Hilfe des öffentlichen Schlüssels des Nutzers verifizieren, der jedermann zugänglich ist. Natürlich erfordert die Signaturverifikation, dass die über die signierten TLS Handshake Nachrichten konstruierte Nachricht dem Verifizierer zur Verfügung steht. Die `CertificateVerify` Nachricht wird für die Verifikation benutzt, ob der Sender der richtige Eigentümer des Kundenzertifikates ist.

Die Struktur dieser Nachricht wird in TLS folgendermaßen definiert:

```
Struct  {
        Signature signature;
        } CertificateVerify;
```

Der Signaturtyp ist folgendermaßen definiert :

```
select (SignatureAlgorithm)
    {   case anonymous: struct { };
        case rsa:
            digitally-signed struct {
            opaque md5_hash[16];
```

```
                opaque sha_hash[20];
            };
            case dsa:
                digitally-signed struct {
                opaque sha_hash[20];
            };
        } Signature;

    enum { anonymous, rsa, dsa } SignatureAlgorithm;
```

Die Signaturverifikation erfordert die Zerlegung der relevanten TLS Handshake Nachrichten und die Eingabe des Ergebnisses in einen Verifikationsalgorithmus:

```
CertificateVerify.signature.md5_hash
    MD5(handshake_messages);
CertificateVerify.signature.sha_hash
    SHA(handshake_messages);
```

Für unsere Zwecke benötigt man hier das komplette TLS Handshake Protokoll nicht. Einzelheiten entnehmen man [DieAll99]. Für das SSO System werden die signierte Nachricht, die aus den TLS Handshake Nachrichten bis zur `CertificateVerify` Nachricht bestehen, und deren Signatur von der TLS Anwendung extrahiert und vom ISP gespeichert. Dieses Nachricht/Signatur-Paar kann später zu anderen Diensteanbietern gesandt werden, wie zu RSPs, wobei diese Dienstanbieter dann die Nutzerauthentifikation anfordern.

Um diese Signaturen für SSO Zwecke nutzen zu können, benötigen Dienstanbieter einige spezifischen Informationen aus der signierten Nachricht, wenn potentielle Angriffe entdeckt werden sollen. Umgekehrt muss die signierte Nachricht, wenn die TLS Handshake Signatur für SSO genutzt werden soll, einige notwendigen Informationen enthalten, welche die Entdeckung solcher Angriffe erlauben. Glücklicherweise beinhalten die TLS Handshake Nachrichten diese Informationen. Im Folgenden zeigen wir die relevanten Handshake Nachrichten, welche solche wichtigen Informationen enthalten.

`ClientHello`:
Diese Nachricht wird von einem Client abgeschickt, wenn er einen Server kontaktiert.
Die Struktur ist vom TLS Protokoll folgendermaßen gegeben:

```
Struct  {
    ProtocolVersion client_version;
    Random random;
    SessionID session_id;
    CipherSuite cipher_suites;
    CompressionMethod compression_methods;
    }
```

Ein interessanter Teil dieser Struktur ist in dem Wert von `Random` enthalten. Er enthält die Zeit, wenn das TLS Handshake Protokoll ausgeführt wurde. Der Dienstanbieter verifiziert die Zeit, um die Aktualität der Signatur zu prüfen. Der `random` Typ ist definiert, wie folgt:

```
struct  {
        uint32 gmt_unix_time;
        opaque random_bytes[28];
        } Random;
```

Neben der Bereitstellung der Zeit für die Aktualität gibt es eine funktionelle Anforderung, die mit der ClientHello Nachricht gekoppelt ist. Da die Signatur gemäß der übertragenen Cipher Folge generiert ist, fordert der Verifikator die Information, welcher Mechanismus benutzt werden muss, um das TLS cipher Handshake Nachricht/Signatur-Paar zu.

`ServerCertificate`:
Wie der Name sagt, enthält diese Nachricht das Zertifikat des Servers. Ein Dienstanbieter kann den Inhalt verifizieren, um zu sehen, ob der Nutzer wirklich vom Eigentümer des Zertifikates authentifiziert war. Die Struktur dieser Nachricht ist im TLS handshake Protokoll folgendermaßen definiert:

```
struct {
        ASN.1Cert certificate_list;
        } Certificate;
```

`ClientCertificate`:
Diese Nachricht enthält das Clientzertifikat. Seine Struktur ist die gleiche wie die des

`ServerCertificate`’s. Das Clientzertifikat ist sicher eine essentielle Information, welche vom ISP und von anderen Authentifizierungsinstanzen angefordert werden.

Wie aufgezeigt wurde, enthalten die TLS Handshake Protokoll die notwendigen Informationen, welche die Entdeckung von Angriffe erlauben. Diese signierten TLS Handshake Nachrichten und die Signaturen auf ihnen können, wie sie aus der TLS Anwendung extrahiert sind, vom ISP als zwei binäre codierte oder 64 Bit basierte Strings gespeichert werden. Im Falle des SSO sendet der ISP zwei Strings an andere relevante Dienstanbieter, indem er den SAML Standard benutzt.

8.4.2 Generation von Authentifikationsversicherungen in SAML

Mit der verifizierbaren Authentifikation aus dem TLS Handshake Protokoll muss der ISP eine Authentifikationsversicherung generieren, die an die Diensteanbieter für SSO gesandt werden muss. SAML definiert eine Authentifikationsversicherung, die in diesem Fall als Anweisung dient. Eine solche Anweisung ist in Abbildung 160 angegeben. Sie

wird vom kontaktierten ISP zum RSP des Nutzers gesandt, damit die notwendige Authentifikationsinformation bereitgestellt werden kann.

```
<saml:Assertion>
      <saml:AuthenticationStatement
            AuthenticationMethod="…URI…"
            AuthenticationInstant="2001-12-03T10:02:00Z">
            <saml:Subject>
                  <saml:NameIdentifier
                  SecurityDomain="smithco.com"
                  Name="joeuser" />
                  <saml:SubjectConfirmation>
                        <saml:ConfirmationMethod>
                        Cipher Suite
                        </saml:ConfirmationMethod>
                        <saml:SubjectConfirmationData>
                         Signed TLS handshake messages (base64)
                         tag
                         Signature (base64)
                        </saml:SubjectConfirmationData>
                  </saml:SubjectConfirmation>
            </saml:Subject>
      </saml:AuthenticationStatement>
</saml:Assertion>
```

Abbildung 160: SAML Authentifikationsversicherung

Die `ConfirmationMethod` ist eine spezifische Methode, welche es der trauenden Partei erlaubt, sicher zu sein, dass Nachricht von der Instanz kommt, die mit dem Subjekt in der Anweisung korrespondiert [SAML02a]. Die `SubjectConfirmationData` ist in SAML als String definiert. Als Folge muss er konvertiert werden, wenn er in einem ungeeigneten Format bereitgestellt ist.

Angenommen, die signierten Authentifikationsversicherungen und der Signaturwert sind in den kontaktierten ISP in das `SubjectConfirmationData` Element eingefügt, d.h. der erste String mit den Authentifikationsversicherungen und der zweite String mit seinem signierten zerlegten Wert sind in einem String in der SAML Authentifikationsversicherung gespeichert, kann eine Marke eingefügt werden, sie zu trennen. Wenn ein Dienstanbieter einen solchen String erhält, kann er dann leichter zwischen der Information und dem signierten Hashwert differenzieren. Da im TLS Handshake Protokoll die signierten TLS Handshake Nachrichten und der Signaturwert binär vorliegen, können sie nicht unmittelbar im TLS Handshake Protokoll benutzt werden. Deshalb werden beide Strings ins „base64“-Format konvertiert. Um die saubere Separation der beiden Strings zu garantieren, wird als Separationsmarke ein Zeichen gewählt, das nicht im „base64“-Alphabet enthalten ist, z.B. „#”.

Der Empfänger der SAML Versicherung identifiziert die beiden „base64“ kotierten String im `SubjectConfirmationData` Element mit Hilfe der eingefügten Marke und dekodiert dann die „base64“ Strings in Binärstrings. Danach kann er die Signatur mit dem Mechanismus aus der `ConfirmationMethod` Element verifizieren.

Ein anderer wichtiger Aspekt liegt darin, dass SAML Authentifikationsversicherungen von ihrem Erzeuger mit XML Signaturen signiert werden können. Dies versichert die Integrität und den Ursprung der Authentifikationsversicherung [SAML02a]. Wie in der hier vorgeschlagenen Lösung enthält die Versicherung eine vom Nutzer signierte Information, welche dem ISP enthüllt, mit wem der Nutzer korrespondiert. Diese Information kann vom Empfänger der Authentifikationsversicherung, wie den RSPs, benutzt werden, das Zertifikat des ISP aus der vom Nutzer signierten Information und die Signatur des ISP als abzugleichen. Dies erlaubt die Verifikation, dass der Absenden der Versicherung und der vom Nutzer kontaktierte ISP dieselbe Partei sind.

SAML Nachrichten können an jede Partei über das „Simple Object Access Protocol“ (SOAP), welches Mechanismen für die Definition von Nachrichten in XML und für deren Versendung durch HTTP beschreibt, versandt werden [SOAP1.1]. SOAP besteht aus drei Hauptteilen: einer Hülle, Kopfdaten und einem Nachrichtenkörper (an envelope, header data and a message body). Eine wichtige Eigenschaft von SOAP Nachrichten ist, dass sie grundsätzlich Einwegnachrichten sind, d.h. die Nachrichten können an einen Empfänger gesendet werden, ohne dass dieser antworten muss. Nichtsdestotrotz muss der Empfänger als Komponente des Dienstanbieters in der Lage sein, das SOAP zu verstehen und die Nachricht zu interpretieren. SOAP Nachrichten können dazu benutzt werden, “request/response”-Implementationen zu realisieren [SOAP1.1]. Die zugehörigen SAML Nachrichten werden über SOAP transportiert. Der Absender fügt die SAML Nachricht in den SOAP Nachrichtenkörper ein und sendet diesen zum Empfänger. Der Empfänger extrahiert die SAML Nachricht aus dem SOAP Nachrichtenkörper und verarbeitet sie. Wenn die SAML Nachricht eine Antwort fordert, dann sendet der Empfänger eine solche in einem SOAP Nachrichtenkörper zurück.

In der vorliegenden Lösung kann der ISP die SAML Versicherung in einen SOAP Nachrichtenkörper einfügen in diesen an einen Dienstanbieter z.B. einen RSP senden. Der RSP kann die SAML Versicherung aus dem SOAP Nachrichtenkörper extrahieren und ggf. weiter gefordert Prozeduren ausführen. Ähnlich kann der ISP SOAP Nachrichten an andere Dienstanbieter erforderlichenfalls versenden. Auf diese Weise führt das Nutzergerät nur eine Authentifizierung durch, nämlich zum kontaktierten ISP und die notwendige Authentifikationsinformation wird vom IPS bereitgestellt.

8.5 Vorteile der SSO Lösung

In diesem Abschnitt werden die Vorteile der SSO Lösung zusammengefasst, sowie ihre Nachteile diskutiert. In der SSO Lösung wird der Nutzer vom ISP authentifiziert und der ISP versorgt andere Parteien, wie RSPs, mit einer verifizierbaren Authentifikationsinformation über den Nutzer. Als Folge kann ein ISP keine schädliche

Erklärung abgeben, einen speziellen Nutzer authentifiziert zu haben, ohne dies in Wirklichkeit auch durchgeführt zu haben.

Da die Authentifikation auf der öffentlichen Schlüsselkryptographie und Zertifikaten beruht, unterstützt die Lösung Praktikabilität und Sicherheit eines Nutzers, da es zwischen den Parteien keine im Voraus festgelegten Geheimnisse gibt. Dies ist ein besonderer Vorteil für Erreichbarkeitsszenarien mit adhoc Beziehungen, bei denen nicht angenommen werden kann, dass Nutzer und der kontaktierter ISP als Authentifikator bzw. der ISP und der RSP sich vertrauen. Ein Ziel der SSO Lösung ist die Unterstützung der schnellen Authentifikation, welche nicht nur auf die anfängliche Entität Authentifikation durch den ISP, sondern auch auf die Authentifikation der anderen Diensteanbieter gerichtet ist. Die vorliegende Lösung löst diesen Aspekt durch die Reduzierung der Anzahl Interaktionen für SSO und durch Verschieben der SSO Verantwortlichkeit weg von Komponenten, die Engpässe hervorrufen können. Da mobile Geräte und die Kommunikation mit diesen Engpässe hervorrufen können, folgt unser Ansatz einer strengeren Definition des SSO, wobei nicht nur die Zahl der Nutzerinteraktionen sondern auch die Interaktionen der Nutzergeräte reduziert sind.

Die Kryptographie öffentlicher Schlüssel, d.h. die Generierung und die Verifikation digitaler Signaturen, erfordert umfangreiche Computerkapazitäten. Für die Erfüllung der Hochgeschwindigkeitsanforderung ist eine reduzierte Last für solche Computer von Vorteil. Die vorgeschlagene Lösung nutzt mehrfach die Ergebnisse derartiger Berechnungen, ohne den Sicherheitsstand zu erniedrigen, indem die signierte Nachricht und ihre Signatur vom TLS Handshake Protokoll extrahiert und für die Erzeugung der verifizierbaren Authentifikationsversicherungen erneut benutzt werden.

Man könnte nun sagen, dass die verifizierbaren Authentifikationsversicherungen aus den Nachrichten des TLS Handshake Protokolls und der Nutzersignatur einen überflüssigen Overhead und eine Verschwendung von Kapazität seien. Dem ist entgegenzuhalten, dass die Signatur für die Authentifikation im TLS Protokoll genau so definiert ist, und es gibt keine andere Möglichkeit, wenn die existierenden Sicherheitsstandards die Grundlage sein sollen. Wenn einige Teile in den Handshake Nachrichten weggelassen würden, kann die Signatur nicht mehr verifiziert werden mit der Folge des vollständigen Verlustes der Sicherheit. Das Ausnutzen der Signatur des anfänglichen Authentifikationsprotokolls und eine Reduzierung des Umfangs der auszutauschenden Daten würde eine Modifikation des TLS Standards erfordern.

Um die Qualität der vorliegenden Lösung beurteilen zu können, wurde das Datenaufkommen gemessen, welches in typischen TLS Handshakes auf dem TCP Niveau ausgetauscht wird. Das Ergebnis zeigte, dass der größte Umfang der ausgetauschten TLS Handshake Daten vom ausgetauschten öffentlichen Schlüssel und vom Zertifikat stammen. Beide sind notwendig für die Verifikation der Signatur. In der überwiegenden Anzahl der Fälle beanspruchen diese Daten mehr als 90 % des kompletten Handshakes. Da diese Daten immer benötigt werden, ist die Verschwendung von Platz viel geringer als vermutet.

Weiter ist der Aufwand, der bei einer Implementierung der Lösung entsteht, zu beachten. Da die Lösung auf Standardprotokollen basiert, gibt es auf Seiten des Nutzers oder Client keine zusätzlichen Implementationsanforderungen. Auf der Seite der ISP oder Server muss die signierte Nachricht und die Signatur aus der TLS Implementation extrahiert werden. Der Implementationsaufwand sollte klein sein. Er hängt von der vorhandenen TLS Implementation ab. Weiter ist eine Komponente für die Kreation der SAML Nachrichten zu entwickeln. Die Kosten hierfür sind niedrig.[32] Weiter ist eine Komponente für den Transfer der SSO Nachrichten erforderlich. Die Kosten hierfür sind ebenfalls niedrig[32], wie die für die Empfängerkomponente auf Seiten der RSP. Weiter muss der RSP die Authentifikationsversicherungen und die eingebettete digitale Signatur des Nutzers verifizieren können. Der RSP muss den kryptographischen Algorithmus unterstützen, der von der TLS Implementation für die Erzeugung der Nutzersignatur benutzt wird. Die Implementation hierfür ist umfangreich[32]. Die Kosten hierfür können durch die erneute Nutzung von vorhandenem Code der TLS Implementation reduziert werden. Die Implementation lässt sich auf einen kleinen Teil von existierenden Cipher Folgen reduzieren[32].

8.6 Konklusion

In diesem Abschnitt wurde eine SSO Lösung für das Roaming VPN Zugangsdienst Geschäftsmodell entwickelt. Die Lösung ist allgemein genug, um auch auf die anderen Geschäftsmodelle aus Kapitel 3 angewandt werden zu können. Da die Authentifizierungslösung aus Kapitel 6 auf asymmetrischer Kryptographie und Zertifikaten beruht, nutzt auch die SSO Lösung diese Mechanismen. Die Lösung wird dem hier zugrunde gelegten Roaming Szenaro gerecht, da nicht nur die Anzahl der Nutzerinteraktionen, sondern auch die Interaktionen bei der Authentifizierung und der Arbeitsaufwand für das Endgerät des Nutzers gering sind. Die vorgeschlagene Lösung erlaubt SSO auf Basis digitaler Signaturen, welche innerhalb des TLS Handshake-Protokolls zur Authentifizerung erzeugt werden. In dem hier vorgeschlagenen SSO System werden diese Signaturen dann bei der Erzeugung von auf dem SAML Standard beruhenden „Authentication Assertions“ wiederverwendet.

Im nächsten Kapitel wird die Implementierung einer sicheren beidseitigen zertifikatebasierten Authentifizierung auf Basis des TLS-Handshake-Protokolls vorgenommen.

[32] Die Abschätzung des Aufwandes bzw. der Kosten erfolgte im Rahmen einer Kooperation mit den Fujitsu Laboratories Europe [FLL03]

9 Implementierung

Da eine sichere beidseitige Authentifizierung der Kern der in dieser Arbeit enwtickelten Architektur über alle Modelle und Fälle hinweg ist, wird eine zertifikatebasierte sichere beidseitige Authentifizierung hier implementiert. Zum einen wird die Modifikation des TLS/SSL Handshakes implementiert[33], wie im Abschnitt 9.1 beschrieben. Zum anderen wird die beidseitige Authentifikation mit Port basierter Zugangskontrolle nach 802.1x für WLAN und WIMAX unter Linux implementiert, wie in Abschnitt 9.2 beschrieben.

9.1 Authentifikation mit modifiziertem Handshake

Für die Erstellung des ersten Prototyps wurde Visual Studio 2002 Version 7.0.9466 als Entwicklungsumgebung eingesetzt. Die genaue Konfiguration der Entwicklungsumgebung ist im Anhang in Kapitel 15.1 beschrieben. Die Quelldateien für den in diesem Kapitel dargestellten Prototyp befinden sich auf der CD im Anhang. Eine Übersicht über diese gibt Anhang 15.4.

9.1.1 Beschreibung der Funktionalität und Implementation des Prototyps

Nun werden die Funktionalität des Prototyps und die Art der Implementierung beschrieben. Der Prototyp demonstriert eine auf Zertifikaten basierende beidseitige Authentifizierung zwischen einem Client und einem Server, bei welcher die Überprüfung der Zertifikate zu PKI Servern delegiert wird. Dies wird erreicht durch die Erweiterung und Modifizierung des SSL(TLS)-Handshake zwischen den Authentifizierungs-Endpunkten. Abbildung 161 gibt einen Überblick über die Funktionsweise des erweiterten SSL(TLS)-Handschlags und der Art, wie er implementiert ist.

Sowohl der SSL(TLS)-Client als auch der SSL(TLS) Server kennen zwei Betriebsmodi. Im Mode 0 verhält sich der Client wie ein herkömmlicher SSL(TLS)-Client. Im Mode 1 dagegen - dem extended mode – sendet der Client in einer erweiterten „Client_Hello“ Nachricht die Adresse oder den Hostname eines PKI Servers, dem er vertraut.

Wenn auf der anderen Seite der SSL(TLS) Server in Mode 0 ist, dann akzeptiert er sowohl die Kommunikation mit normalen SSL(TLS)-Clients als auch mit erweiterten SSL(TLS)-clients. Wenn der SSL(TLS) Server in Mode 1 ist, dann akzeptiert er erweiterte SSL(TLS)-Clients. Der Server extrahiert die Information über den Mode, indem der Client sich befindet, aus der erhaltenen „Client_hello“ Nachricht. Um zu erkennen, ob der Handshake erweitert ist

[33]Im Rahmen einer Kooperation mit den Fujitsu Laboratories Europe

oder nicht sucht der Server nach einem Tag der zusammen mit der Adresse des PKI Servers innerhalb der Client_hello Nachricht vom Client geschickt wird. Wenn die erhaltene Client_Hello Nachricht erweitert ist, extrahiert der SSL(TLS) Server die IP Adresse bzw. den Hostname des PKI Servers und fährt fort als ereiterter SSL(TLS) Server.
Er sendet einen Zertifikatsüberprüfungs- Request zu der gegebenen Adresse des PKI Servers. Abbildung 161 illustriert diese Prozedur im Detail. Der SSL(TLS) Server startet ein Java Programm mit Namen “Testshell” als „child“-Prozess. Dieses Programm arbeitet als ein Mediator. Es sendet einen Request zum PKI Server und erhält auch die Antwort von diesem, worauf es diese Antwort dem SSL(TLS) Server zu Verfügung stellt. Wenn der SSL(TLS) Server keine Antwort erhält wird der Handshake abgebrochen.
Der SSL(TLS) Server erzeugt eine neue “validation message”, welche das Ergebnis der vom PKI Server durchgeführten Überprüfung enthält und sendet diese zum Client. Der Client erwartet das Ergebnis und überprüft den Inhalt sowie die Urheberschaft. Zu diesem Zweck startet der Client ein Java-Programm namens “testshell2” als „child“-Prozess. Dieser Prozess überprüft die Signatur und interpretiert den Validationsstatus. Wenn die Signatur gültig ist und das Ergebnis sagt, dass das überprüfte Zertifikat gültig ist, dann wird der Handschlag fortgesetzt, wenn nicht, wird der Handschlag terminiert.

Wenn Client uns Server im Mode 0 arbeiten, wickeln sie das Protokoll wie in Kapitel 2.3.3 beschrieben ab.

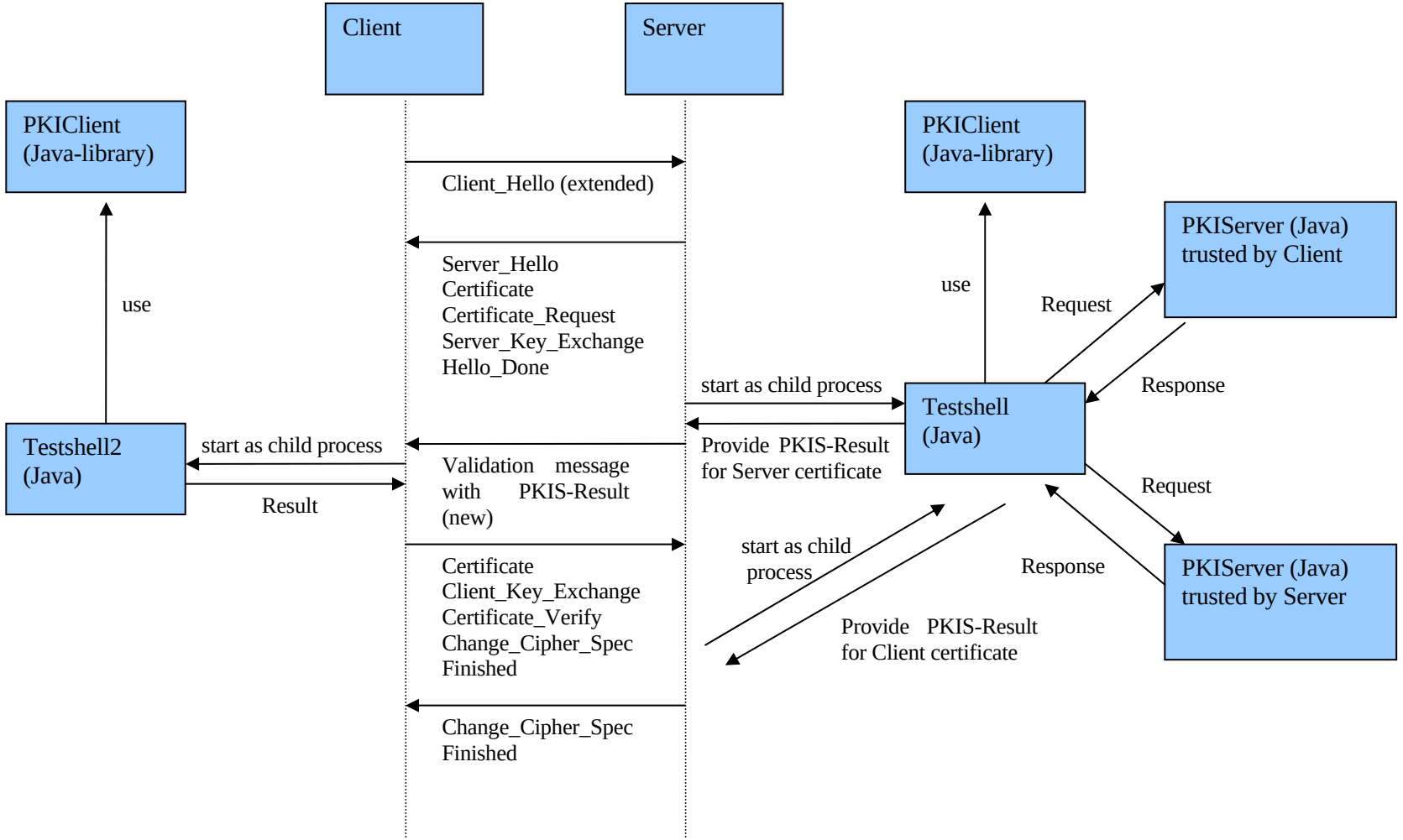


Abbildung 161: Überblick über die Implementation

Im Fall der Fortführung des Handschlags sendet der SSL(TLS)-Client sein Zertifikat zum SSL(TLS) Server. Der Server sendet wieder einen Request zu einem PKI Server, dem er vertraut und erwartet die Antwort. Zu diesem Zweck wird wieder das Mediator-Programm “Testshell” verwendet. Der Unterscheid im Vergleich zum ersten Aufruf ist, dass der PKI-Client die Unterschrift des PKI Servers überprüfen kann für den SSL(TLS) Server. Wenn der Status des Zertifikates des SSL(TLS)-clients “not valid” ist, dann endet der Handshake. Wenn er “valid” ist, dann läuft der Handshake ganz normal weiter.

9.1.2 Test

In der Testphase wird untersucht, ob die Kommunikation zwischen dem SSL(TLS)-Client und Server sowie die Kommunikation mit dem PKI Server so funktioniert, wie sie es soll. Zu diesem Zweck werden zunächst alle Komponenten auf einem PC installiert und die Kommunikation getestet. Darauf wurden sie auf unterschiedlichen vernetzten PCs installiert und der korrekte Ablauf des Handshakes wurde wiederum getestet. All Tests ergaben positive Ergebnisse. Wenn unterschiedliche PCs verwendet werden, müssen die Hostnamen und IP-Adressen in der Datei common.h entsprechend geändert werden. Zusätzlich zu den Zertifikaten im Verzeichnis “ClientServer\certs\” werden CAs, die der zugrunde liegenden Roaming Architektur aus dieser Arbeit entsprechen, aufgesetzt. Die Befehle, die notwendig sind, um CAs und Zertifikate zu erstellen sind im Detail in http://www.openssl.org/docs/apps/openssl.html beschrieben. Nichtsdestotrotz gibt der Anhang 15.2 einen Überblick über die Erzeugung von CAs und Zertifikaten mit OpenSSL.

Zum Testen werden 3 CAs entsprechend der in Kapitel 6 beschriebenen Lösung aufgesetzt: CAofRSP1, CAofRSP2 and CAofEnterprise. Die RSP1 und RSP2 CAs zertifizieren sich gegenseitig über Kreuz, wie in Abschnitt 6.1.3 beschrieben. Die Unternehmens-CA wird von der CA des RSP1 zertifiziert. Die CA des Unternehmens stellt die Zertifikate user1E und user2E für zwei Clients aus. Die CA von RSP1 stellt die Zertifikate user1RSP1 und user2RSP1 für 2 Client aus und ein Server-Zertifikat. Die CA of RSP2 stellt die Zertifikate user1RSP2 und user2RSP2 für 2 Client aus und ebenfalls ein Server-Zertifikat. Ein sql-script “NSITESTDB.sql”, um die PKI Server-Datenbank zu erstellen, auf denen diese Teste basieren, befindet sich in Verzeichnis CAs. Wie man CAs mit OpenSSL aufsetzt und Zertifikate und CRLs erstellt, wird im Anhang in Kapitel 15.2 genauer beschrieben.

Die Tabelle 10 gibt eine Synopsis der Ergebnisse der Tests der in dargestellten Implementierung unter Verwendung der verschiedenen Zertifikate und Strategien.

Tabelle 10: Tests der Implementierung

Client		Server		Connection		
Client CERTFILE	Client CAFILE	Server CERTFILE	Server CAFILE	not ext.	1.2.3.4	1.2.3.9
client.pem	rootcert.pem	server.pem	rootcert.pem	1	1	1
user1RSP1.pem	rootcert.pem	server.pem	(RSP1)CAcert.pem	1	1	1
user2RSP1.pem	rootcert.pem	server.pem	(RSP1)CAcert.pem	1	0(CI)	1
user1RSP2.pem	rootcert.pem	server.pem	(RSP2)CAcert.pem	1	0(CU)	1
user2RSP2.pem	rootcert.pem	server.pem	(RSP2)CAcert.pem	1	0(CU)	1
userE1.pem	rootcert.pem	server.pem	(Ent.)CAcert.pem	1	1	1
userE2.pem	rootcert.pem	server.pem	(Ent.)CAcert.pem	1	0(CI)	1
client.pem	(RSP1)CAcert.pem	(RSP1)server.pem	rootcert.pem	1	1	1
user1RSP1.pem	(RSP1)CAcert.pem	(RSP1)server.pem	(RSP1)CAcert.pem	1	1	1
user2RSP1.pem	(RSP1)CAcert.pem	(RSP1)server.pem	(RSP1)CAcert.pem	1	0 (CI)	1
user1RSP2.pem	(RSP1)CAcert.pem	(RSP1)server.pem	(RSP2)CAcert.pem	1	0(CU)	1
user2RSP2.pem	(RSP1)CAcert.pem	(RSP1)server.pem	(RSP2)CAcert.pem	1	0(CU)	1
user1E.pem	(RSP1)CAcert.pem	(RSP1)server.pem	(Ent.)CAcert.pem	1	1	1
user2E.pem	(RSP1)CAcert.pem	(RSP1)server.pem	(Ent.)CAcert.pem	1	0 (CI)	1
client.pem	(RSP2)CAcert.pem	(RSP2)server.pem	rootcert.pem	1	0 (SU)	1
user1RSP1.pem	(RSP2)CAcert.pem	(RSP2)server.pem	(RSP1)CAcert.pem	1	0 (SU)	1
user2RSP1.pem	(RSP2)CAcert.pem	(RSP2)server.pem	(RSP1)CAcert.pem	1	0 (SU)	1
user1RSP2.pem	(RSP2)CAcert.pem	(RSP2)server.pem	(RSP2)CAcert.pem	1	0 (SU)	1
user2RSP2.pem	(RSP2)CAcert.pem	(RSP2)server.pem	(RSP2)CAcert.pem	1	0 (SU)	1
userE1.pem	(RSP2)CAcert.pem	(RSP2)server.pem	(Ent.)CAcert.pem	1	0 (SU)	1
userE2.pem	(RSP2)CAcert.pem	(RSP2)server.pem	(Ent.)CAcert.pem	1	0 (SU)	1

Die ersten beiden Spalten namens „Client“ und „Server“ geben jeweils die CERTFILEs und CAFILEs an, welche in den client.c and server.c Dateien definiert werden müssen. Die “Connection”-Spalten zeigen, ob der Handshake erfolgreich war (1) bei Einsatz der in den zugehörigen Zeilen angegebenen Zertifikate oder nicht (0). Wenn nicht, dann ist der Grund für den Abbruch des Handshakes in Klammern angegeben. CU bedeutet, dass das Zertifikat des Client unbekannt ist, SU bedeutet, dass das Zertifikat des Servers unbekannt ist und CI meint, dass das Zertifikat des Clients ungültig ist.

9.1.3 Einsatzmöglichkeit des Prototyps

Der Prototyp besteht aus einem SSL(TLS)-Client und Server, welche einen Handshake basierend auf OpenSSL ausführen. Der Client und der Server kennen zwei Modi. Im erweiterten Modus überprüft ein PKI Server die Zertifikate innerhalb des Ablaufes des Handshakes. Die Abbildung 162 illustriert ein Einsatzszenario für den Prototyp.

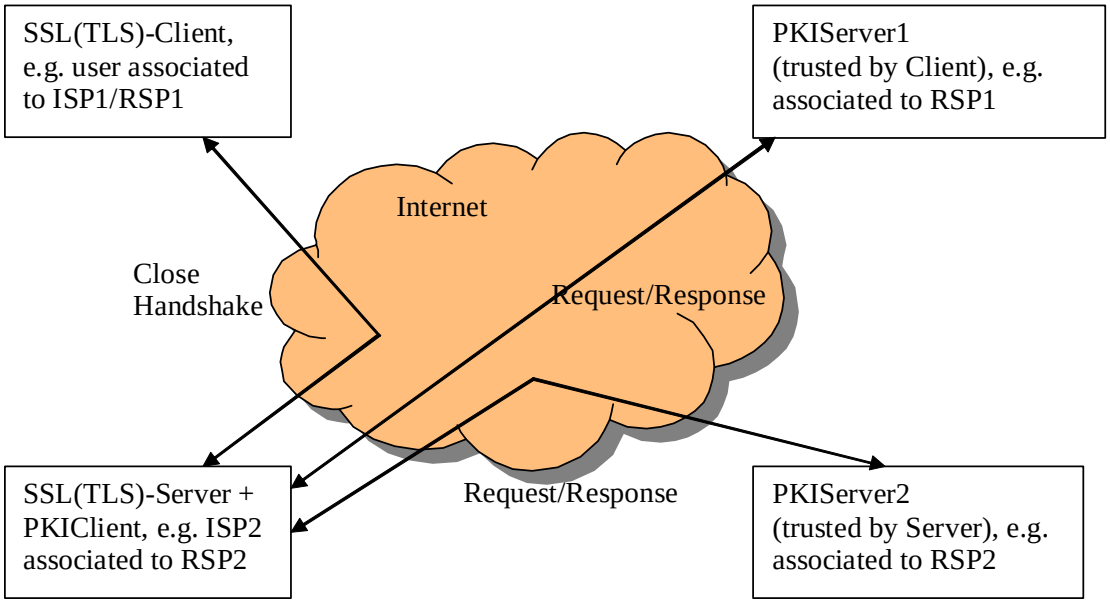


Abbildung 162: Szenario für den Einsatz des Prototyps

Die Entscheidung, ob der "extended mode" verwendet wird oder nicht und ob TLS oder SSL verwendet werden basiert auf der Erzeugung des SSL-context-objects.
Der Client und der Server erzeugen ein context-objects ctx. Dies geschieht durch den Befehl SSL_CTX_new, z.B. so: `ctx = SSL_CTX_new(SSLv3_method( 1 ));` zur Erzeugung eines SSL basierten Objektes oder mit `ctx = SSL_CTX_new(TLSv1_method( 1 ));` zur Erzeugung eines TLS basierten Objektes.

Die SSLv3_method und die TLSv1_method Funktionen sind erweitert auf eine solche Weise, dass sie jetzt einen integer Parameter akzeptieren, wenn sie aufgerufen werden. Falls der Parameter „0" ist, ist der Mode "not extended" und falls der Parameter auf „1" gesetzt ist, wird der „extended mode" aktiviert.

Im Anhang in Kapitel 15.3 sind Screenshots von Instanzen jeweils eines Clients und eines Servers dieses Prototypen, welche zusammen das Handshake Protokoll miteinander abgewickelt haben. In Kapitel 15.3.1 zeigen Abbildung 171 und Abbildung 172 je einen Client und einen Server, welche im nicht erweiterten Modus das TLS Protokoll miteinander abgewickelt haben. Kapitel 15.3.2 zeigt in Abbildung 173 und Abbildung 174 je einen Client und einen Server, welche das TLS Handshake Protokoll erfolgreich im erweiterten Modus miteinander abgewickelt haben. Um in diesem Dokument zu demonstrieren, dass auch der SSLv3-Handshake genauso wie der TLS Handshake durchgeführt werden kann, zeigen die Abbildung 175, Abbildung 176, Abbildung 177 und Abbildung 178 in Kapitel 15.3.3 je eine Instanz eines erweiterten und eines nicht erweiterten SSL-Client und Servers nach einem erfolgreich durchgeführten Handshake.

9.2 Spezifische Lösung für WLAN und WiMAX

Es wurde im Rahmen dieser Arbeit ein zweiter Prototyp für die Authentifikation entwickelt. Im Unterschied zum ersten Prototyp ist dieser zweite nicht nur eine Implementierung des modifizierten Handshakes, sondern eine spezifisch für WLAN, WiMAX und Ethernet implementierte Authentifikationslösung nach IEEE 802.1x. Die folgende Abbildung 163 zeigt die einzelnen Hardwarekomponenten mit der zugehörigen für den Aufbau relevanten Software und den Schnittstellen.

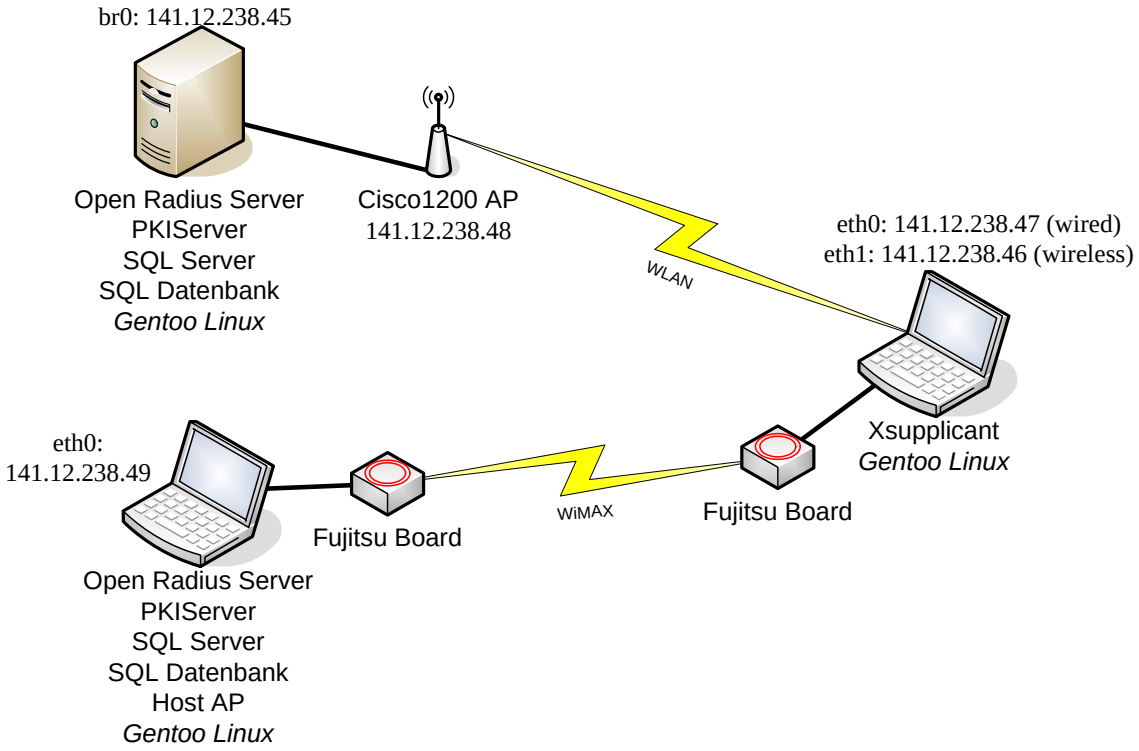


Abbildung 163: Versuchsaufbau

Das genaue Setup des Netzes bzw. die Installation und Konfiguration der Komponenten wird im Anhang 15.4 beschrieben.

Für die Authentifikation über WLAN wird ein Dell Server mit angeschlossenem Cisco AP verwendet. Auf dem Server werden ein „open source“ Radius Server [http://www.xs4all.nl/evbergen/openradius/] und ein PKI Server[34] installiert. Der PKI Server benötigt eine SQL-Datenbank mit den für ihn notwendigen Daten, um arbeiten zu können. Dementsprechend muss noch ein SQL Server und eine passende SQL-Datenbank installiert werden. Alles läuft unter dem Betriebssystem Gentoo Linux ab. Der Client, welcher sich gegenüber dem Server authentisiert ist ein Laptop, auf dem ein „open scource“ Supplikant [http://open1x.sourceforge.net/] installiert ist. Auch der Client läuft

[34] Entwickelt am Fraunhofer SIT in Darmstadt

unter Gentoo Linux. Er kommuniziert mit dem Server über die eth1 Schnittstelle. Der Ablauf der Authentifikation ist in folgender Abbildung 164 dargestellt.

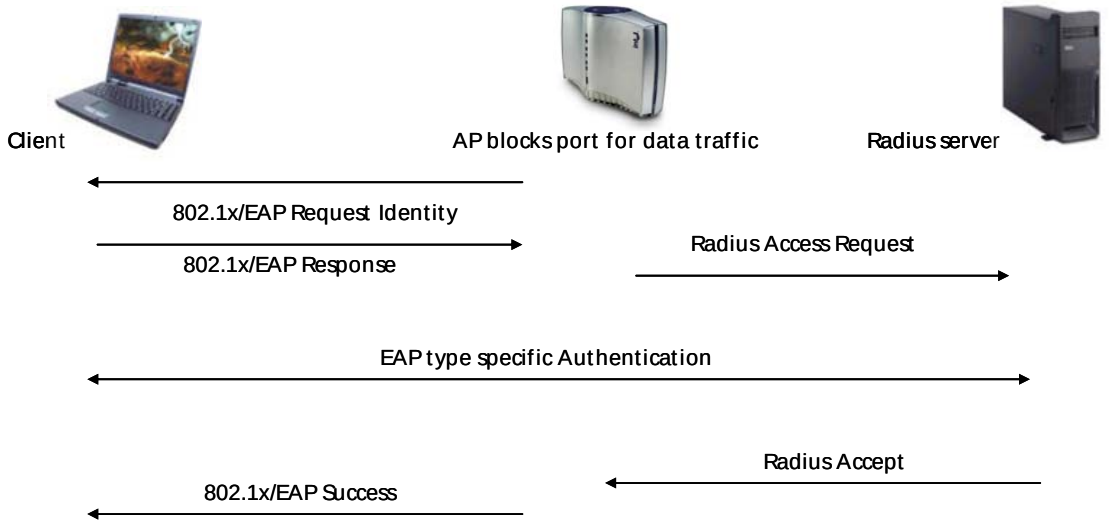


Abbildung 164: Authentifikation nach 802.1x

Das EAP Protokoll wird in RFC2284 genau beschrieben. Die Authentifizierung nach IEEE 802.1x ist „port“-basiert. Das bedeutet, dass der Client zunächst keinen Zugang hinter den AP hat bis dieser den entsprechenden Port freigeschaltet hat. Als EAP-Typ wird TLS verwendet. Der hierfür modifizierte TLS-Handshake ist in folgender Abbildung 165 dargestellt. Die Client_Hello Nachricht wird hier wie schon im vorigen Abschnitt 9.1 beschrieben modifiziert.

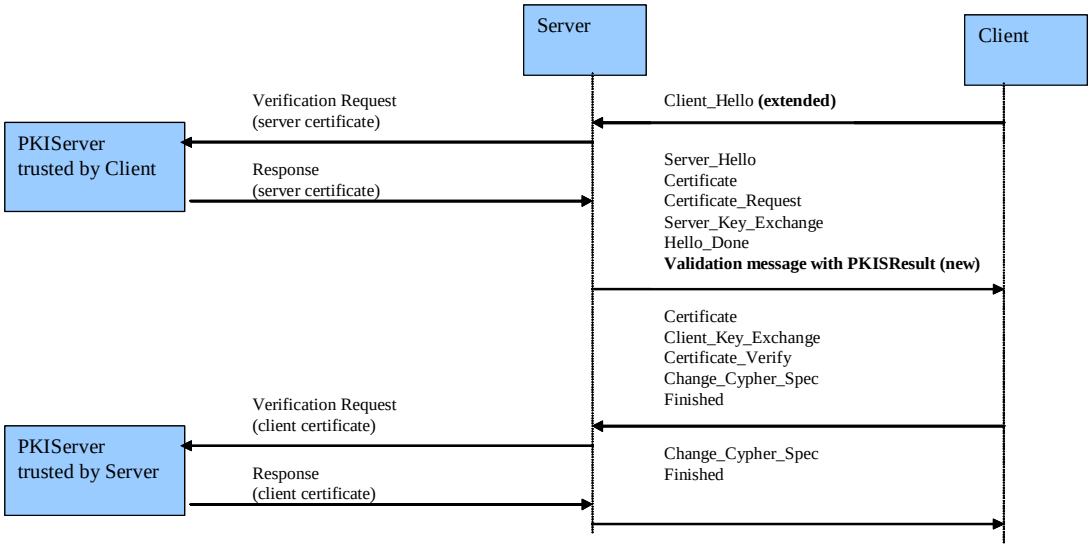


Abbildung 165: Modifizierter TLS Handshake

Im Unterschied zum ersten Prototyp wird hier die Nachricht, welche das Ergebnis der Überprüfung des Serverzertifikates enthält, nicht einzeln geschickt, sondern direkt an die Server_Hello Nachricht angehängt. Ansonsten entspricht der Ablauf des Handshakes dem bereits in Abschnitt 9.1 beschriebenen. Die Sicherheit des SSL/TLS-Standard-Handshake-Protokolls wurde bereits in Abschnitt 2.3.3 dargestellt. Die hier durchgeführten Modifikationen haben keinerlei Einfluss auf die Sicherheit des Handshake-Protokolls an sich, was bedeutet, dass es nach der Modifikation genauso sicher ist wie vorher. Die hinzukommende Kommunikation des TLS Servers mit dem PKI Server bedeutet kein Sicherheitsproblem, da der PKI Server seine Antwort signiert. Dementsprechend wird eine Manipulation der PKI Server-Antwort bemerkt und führt zum Abbruch des Handshakes so, als ob das entsprechende Zertifikat ohne die Einbindung eines PKI Servers vom TLS Server oder TLS-Client nicht für gültig befunden ist.

Bei der Authentifizierung über WiMAX werden auf einem zweiten Laptop wieder der Open Radius Server, der PKI Server und der SQL Server mit passender Datenbank installiert, konfiguriert und die für den erweiterten Handshake notwendigen Änderungen und Erweiterungen implementiert wie schon zuvor. An das „Server“-Laptop und das als Client arbeitende Laptop werden jeweils ein WiMAX Board angeschlossen. Die beiden kommunizieren jeweils über die eth0-Schnittstelle. Da es hier keinen Hardware-AP gibt, wird auf dem „Server“-Laptop ein Host AP installiert.

Die Abbildung 166 zeigt ein Foto des Versuchsaufbaus. Ganz links befindet sich das „Server“-Laptop. Rechts das dunkle „Client“-Laptop. Beide sind mit zwei roten Ethernetkabeln an die WiMAX Boards der Firma Fujitsu angeschlossen. Die beiden Laptops im Hintergrund steuern jeweils das WiMAX Board, hinter dem sie stehen. Die Authentifizierung funktioniert genauso reibungslos wie über WLAN. Der einzige auffällige Unterschied war, dass die bei einem Ping erhaltenen Antwortzeiten bei WiMAX im Rahmen des unten gezeigten Versuchsaufbaus sehr viel höher sind und stark variieren. Sie liegen zwischen 25ms und 45ms im Gegensatz zu WLAN. Bei WLAN liegen die Antwortzeiten bei ca 1,5 ms, wenn das „Client“ Laptop neben dem AP steht. Dies liegt daran, dass die für den Test zu Verfügung stehenden Boards von Software auf angeschlossenen Laptops gesteuerte Prototypen sind. Verkaufsfertige WiMAX APs bzw. Karten werden in der Geschwindigkeit dem WLAN nicht mehr nachstehen.

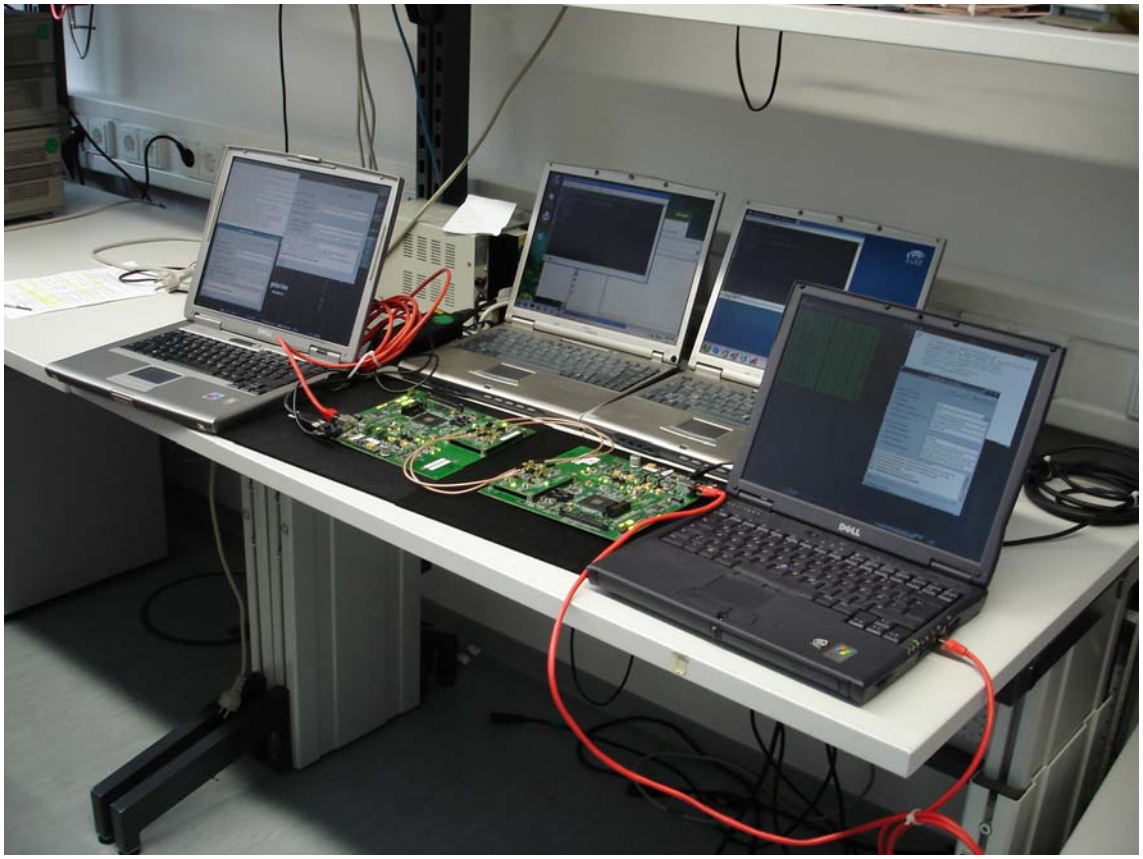


Abbildung 166: Aufbau sichere beidseitige Authentifizierung für WiMAX

Es wird eine graphische Benutzeroberfläche für den Radius Server, den PKI Server und den Client in Java implementiert, über die sich die einzelnen Komponenten bedienen lassen. Sie werden in den folgenden Abschnitten genauer erklärt. In der Abbildung 166 sind sie unscharf zu erkennen.

9.2.1 PKI Server-GUI

Das PKIScoutGUI bietet eine graphische Oberfläche zum Starten und Überwachen des PKI Servers (PKIScout). Das GUI besteht aus Einstellungsmöglichkeiten, Haupt- und Nebenausgabefenstern wie in Abbildung 167 zu sehen ist.

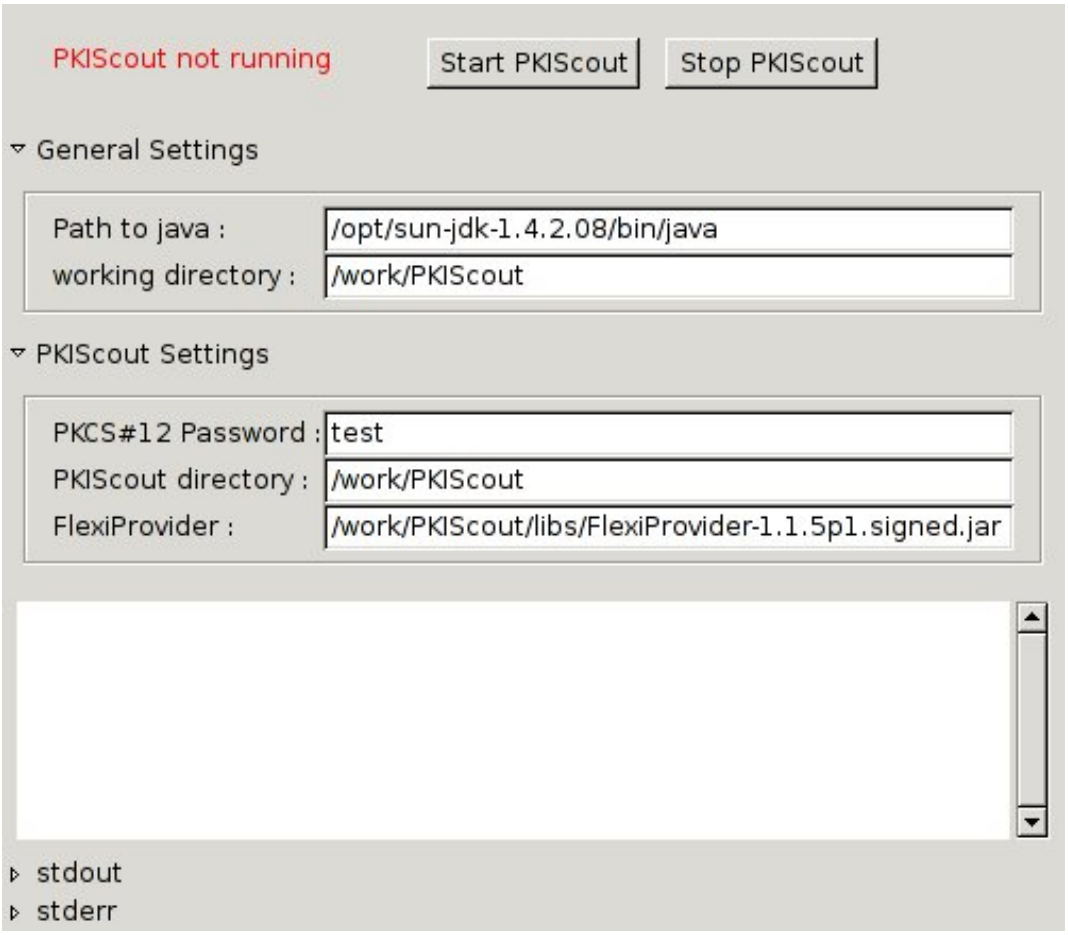


Abbildung 167: PKIScoutGUI

Folgende Einstellungen können vorgenommen werden:

- **Path to java**: Hier muss der absolute Pfad zur ausführbaren Java-Datei angegeben werden. Diese wird dazu benutzt den Server zu starten.
- **working directory**: Das Arbeitsverzeichnis, aus dem der Server gestartet werden soll.
- **PKCS#12 Password**: Falls die PKCS#12 Datei mit einem Passwort geschützt ist, muss dieses hier angegeben werde.
- **PKIScout directory**: Hier wird das Verzeichnis in dem sich das Paket des PKI Servers befindet angegeben.
- **FlexiProvider:** Der PKIScout benötigt den FlexiProvider. Deshalb muss hier der Pfad des FlexiProvider Pakets angegeben werden.

Das **Hauptausgabefenster** enthält die wichtigsten Informationen. Damit lässt sich erkennen, ob der PKIScout die eingehenden Anfragen richtig beantwortet. Weitere Informationen lassen sich über die **stdout**- und **stderr**-Expander einblenden. Auf stdout wird die komplette Ausgabe des PKIScout angezeigt, auf stderr sieht man die Fehler, die von Java auf den Fehlerkanal geschrieben werden, wie z.B. jegliche Art von Exceptions.

Der stderr Kanal ist zur Fehlersuche gedacht, falls der PKIScout nicht richtig startet, z.B. weil eine Datei nicht gefunden werden kann.

9.2.2 RadiusGUI

Das RadiusGUI bietet eine Oberfläche zum Starten und Überwachen des radiusd, der die Authentifizierungsanfragen entgegen nimmt. Das GUI ist unterteilt in eine Einstellungsbereich und einen Ausgabebereich, wie in Abbildung 168 gezeigt. Die genau Erläuterungen zu den Einstellungen finden sich weiter unten.

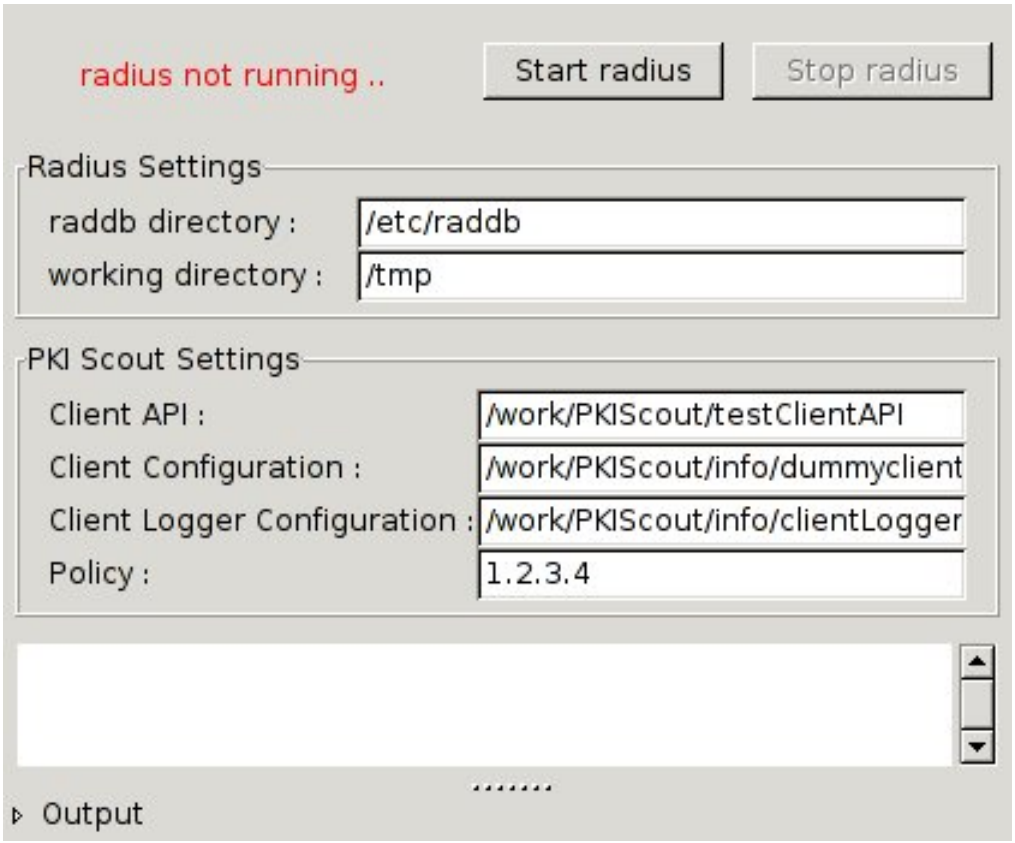


Abbildung 168: RadiusGUI

Es können die folgenden Einstellungen vorgenommen werden:

- **raddb directory:** Hier wird das Verzeichnis angegeben, das die Konfigurationsdateien des FreeRadius enthält.
- **working directory:** Dem Aufruf des radiusd wird dieses Arbeitsverzeichnis übergeben. Zu beachten ist, dass der ausführende Benutzer Schreibrechte in diesem Verzeichnis benötigt, da hier die PKISRESULT.tmp zwischengespeichert wird.
- **Client API**: hier wird die testClientAPI (Das Shell Script das die Testshell aufruft) angegeben.

- **Client Configuration:** Konfigurationsdatei der Client API. Dient als Parameter der TestShell.
- **Client Logger Configuration**: wird ebenfalls als Parameter an die TestShell übergeben und beinhalten Einstellungen für ein eventuellen Logging mit Iog4j.
- **Policy**: Hier muss die Policy angegeben werden, nach der die Verifikation der Zertifikate erfolgt.

Im **Hauptausgabefenster** werden dem Benutzer die wichtigsten Meldungen des radiusd angezeigt. Hier lässt sich der Handshake Schritt für Schritt verfolgen. Für die Fehlersuche lässt sich ein weiteres Fenster namens **output** einblenden, dass die komplette Ausgabe des radiusd beinhaltet.

9.2.3 XsupplicantGUI

Das XsupplicantGUI dient als graphische Oberfläche zum starten des Xsupplicant. Sie ist in Abbildung 169 dargestellt. Eine weitere Möglichkeit ist, den Supplicanten beim Systemstart automatisch als Hintergrundprozess zu starten, dann müssen allerdings beim Extended Handshake die Parameter manuell eingegeben werden.

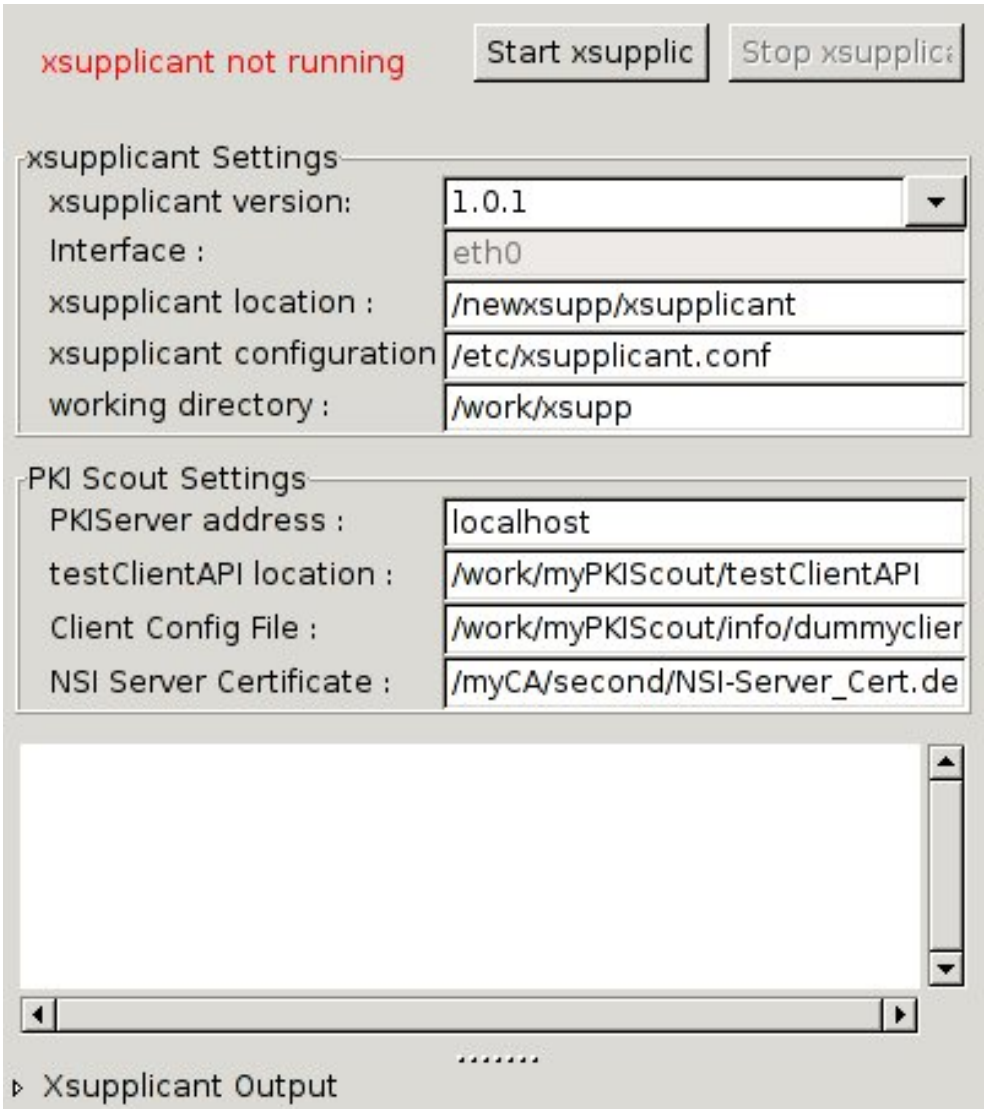


Abbildung 169: XsupplicantGUI

Die folgenden Einstellungen können vorgenommen werden:

- **Xsupplicant Version**: Hier wird die verwendete Version des Xsupplicant ausgewählt. Dies ist nötig, da die Kommandozeilenparameter der beiden Versionen unterschiedlich sind.
- **Interface**: Nur aktiv bei Verwendung von Version 1.2.2. Dort wird das zu verwendende Interface angegeben. Bei Version 1.0.1 geschieht dies in der Konfigurationsdatei xsupplicant.conf.
- **Xsupplicant location**: hier wird der absolute Pfad des Xsupplicant executables angegeben.
- **Xsupplicant configuration**: der absolute Pfad zu der zu verwendenden Konfigurationsdatei. Hierbei ist zu beachten, dass kleine Unterschiede in der Syntax zwischen den Versionen 1.0.1 und 1.2.2 besteht.

- **working directory:** Beim Ausführen des Xsupplicanten wird ein Arbeitsverzeichnis übergeben. Der Benutzer der das XsupplicantGUI startet benötigt Schreibrechte in diesem Verzeichnis.
- **PKI Server address**: Die (IP)-Adresse des PKI Servers, der vom Server verwendet werden soll, um sein Zertifikat prüfen zu lassen. VORSICHT: die Länge der Adresse ist im Moment auf 15 Zeichen beschränkt.
- **testClientAPI location**: Hier wird der Pfad zur testClientAPI (Shell Script, welches die TestShell2 aufruft) angegeben, die vom Client aufgerufen wird, um das Ergebnis des PKI-Servers zu überprüfen.
- **Client config File:** das Konfigurationsfile, das von der TestShell2 verwendet werden soll.
- **NSI Server- Certificate**: Das Zertifikat des PKI Servers, das verwendet werden soll, um die Signatur des Überprüfungsergebnisses des PKI Servers zu überprüfen.

Im **Hauptausgabefenster** werden dem Benutzer die wichtigsten Informationen über den aktuellen Fortschritt der Authentifizierung mitgeteilt. Das ausblendbare **output** Ausgabefenster enthält die komplette Ausgabe des Xsupplicanten und dient dem Debuggen, falls Probleme auftreten. Der Xsupplicant wird mit dem Flag "-d 6" gestartet und gibt die entsprechende Informationsmenge aus.

9.2.4 Einsatzmöglichkeit des Prototyps

Dieser zweite Prototyp könnte zur Zugangskontrolle für ein auf WLAN oder WiMAX basierendes Zugangsnetz verwendet werden. Der Nutzer hat hierbei die Möglichkeit eine sichere beidseitige Authentifizierung abzuwickeln, auch wenn Ihm das Zugangsnetz und dessen Zertifikat völlig unbekannt sind, da - wie oben beschrieben - ein von Ihm als vertrauenswürdig eingestufter PKI Server die Überprüfung des Zertifikates übernehmen kann.

Der Prototyp wurde getestet und die Stabilität der Lösung wird vom Fehlermodell nicht beeinflusst.

10 Ausblick

Das eingangs postulierte Verlangen danach, „always on“ zu sein, wird auch weiterhin zunehmen. In Zukunft werden weiterhin die verschiedensten Zugangsnetztechnologien von den unterschiedlichsten Betreibern eingesetzt werden. Allerdings wird nicht zuletzt wegen Voice over IP (VoIP) im Telefoniebereich und damit auch im Mobilfunkbereich langsam eine Umstellung auf IP-basierte Netze erfolgen, was das Roaming auch mit der hier entwickelten Architektur eher vereinfachen wird, da die im Moment zusätzlich notwendigen Komponenten, wie z. B. ein Home Agent, dann selbstverständlich in allen Architekturen und relevanten Endgeräten vorhanden sein werden. Weitere neue Technologien für drahtlose Kommunikation auf denen Zugangsnetze beruhen können, werden entwickelt werden. Deren Sicherheitseigenschaften und die Möglichkeit diese in die entsprechenden Infrastrukturen einzubinden, muss untersucht werden.

Neue andere aufkommende Zugangstechnologien werden mehr Anwendungen ermöglichen. Die begrenzte Bandbreite des Upload bei UMTS verhindert im Moment noch die Einführung von Anwendungen, welche einen sehr großen Datenaustausch erfordern, wie z. B. Video-Konferenzen mit mehreren Teilnehmern. Die neuen Technologien wie High Speed Downlink Packet Access (HSDPA) und High Speed Uplink Packet Access (HSUPA) sollen in naher Zukunft dieses Manko beheben. Mit HSDPA und HSUPA steigt die Bandbreite weiter auf 14.4 Mbps für den Empfang von Daten und 5.8 Mbps für das Senden von Daten. Beide Technologien benötigen keine neue Hardware, sonder nur Software-Upgrades der UMTS Netze laut Nokia [http://www.heise.de/newsticker/meldung/48226]. Österreich hat gemäß [http://www.telekom-presse.at/channel_mobile/news_23143.html] als erstes Land der Welt flächendeckend den HSDPA Ausbau mit T-Mobile Austria bereits abgeschlossen. Bis dies in allen wichtigen Industrieländern erfolgt ist und auch HSUPA überall zu Verfügung steht, werden noch ein paar Jahre vergehen. Die durch diese Technologien möglichen Anwendungen machen das Roaming an sich wiederum attraktiver und erhöhen den Wert eines Roaming Dienstes bzw. des Roamings, wie in dieser Arbeit vorgeschlagen noch weiter.

Man muss sich darüber im Klaren sein, dass in absehbarer Zeit GSM oder UMTS Netze nicht vollständig verschwinden bzw. durch rein IP-basierte Netze ersetzt werden, da die hier vorhandene Infrastruktur, wenn sie ersetzt wird, zunächst wie üblich, nachdem sie in den führenden Industrieländern von neuen Netzen abgelöst wird, die alten Netze bzw. deren Komponenten und Gerätschaften dann verhältnismäßig preiswert verkauft werden, so dass sie einerseits von kleineren Billiganbietern und andererseits von Anbietern aus strukturschwächeren Ländern mit technologisch niedrigem Niveau wie z. B. aus Afrika aufgekauft und aufgebaut werden, so dass bei einem weltweiten Roaming, der Umgang und die Eigenheiten der unterschiedlichen Zugangsnetztechnologien in absehbarer Zeit weiter berücksichtigt werden müssen. Dem trägt die hier entwickelte Roaming Architektur Rechnung.

In Zukunft wird IPv6 das immer noch weiter verbreitete IPv4 ablösen. Damit wird eine Unterstützung von Mobile IP selbstverständlich von allen Netzkomponenten gewährleistet werden. Das Vorhanden sein des entsprechenden IPv6 Stacks in den Endgeräten und Netzkomponenten wird die hier entwickelte Architektur leichter in der Realität umsetzbar machen und das Roaming prinzipiell vereinfachen bzw. in größerem Maße leichter ohne spezielle zusätzliche Hardware oder Software möglich machen. Dies erfordert allerdings eine lückenlose IPv6 Architektur, da z. B. in Netzen, deren Komponenten auf den Betriebssystemen von Microsoft basieren alle Komponenten auf IPv4 „herunterschalten“, sobald eine einzige Komponente im Netz nicht das IPv6 sondern nur das IPv4 abwickeln kann.
Neue Endgeräte mit begrenzten Ressourcen, wie z. B. in Armbanduhren mit Telefonier- und Organizerfunktion werden immer häufiger über einen IP-Stack verfügen, wodurch sie sich zum Einsatz als Mobile Node im Rahmen eines Roaming Szenarios eignen. Hierdurch können weitere Anforderungen, die aus den Eigenschaften dieser neuen noch nicht existierenden Geräte hervorgehen, entstehen.

Weiter werden schnellere Rechner neue Kryptoverfahren erforderlich machen, da der jetzige Standard AES oder das immer noch sehr häufig eingesetzte 3DES mit vertretbarem Aufwand durch vollständige Enumeration gebrochen werden können.

Eine automatische Netzauswahl, welche nicht nur nach technischen Gesichtspunkten wie zu Verfügung stehender Bandbreite, sondern nach wirtschaftlichen Gesichtspunkten das kostengünstigste Netz auch zwischen unterschiedlichen Betreibern auswählt, wäre sicherlich von großem Vorteil für den Nutzer und ein nächster Schritt zur Erweiterung der hier vorgestellten Architektur.

Der Einsatz von SIM und USIM zur Authentifikation in Mobilfunknetzen könnte einer zertifikatebasierten Authentifikation weichen. Dies würde nur geringe Anpassungen der Infrastruktur der Mobilfunkanbieter benötigen und einen Wechsel zwischen beliebigen IP-basierten Netzten mit der hier entwickelten Roaming Architektur sehr stark vereinfachen. Des Weiteren könnten die Zertifikate gleichzeitig nicht nur zum Netzzugang sondern auch zur Nutzung verschiedenster Internet-Dienste verwendet werden. In wieweit solche Entwicklungen realisiert werden, wird von der Akzeptanz vor allem bei den Mobilfunkbetreibern bestimmt.

11 Schlussbemerkung

In dieser Arbeit wurde eine Architektur für "Seamless Secure Roaming" entwickelt, welche das übergangslose Roaming zwischen heterogenen Netzen aus unterschiedlichen Domänen sicher ermöglicht. Hierfür wurde zunächst ein Überblick über die heutigen Technologien hinsichtlich ihrer möglichen Eignung gegeben und auf ähnliche Projekte hingewiesen. Dann wurden die relevanten Parteien, welche beim Vorgang des Roaming involviert sind, identifiziert und ihre Interessen analysiert. Generische Geschäftsmodelle wurden entwickelt, welche die Beziehungen und Interessen der jeweils beteiligten Parteien berücksichtigen.

In diesen Geschäftsmodellen wird der Wechsel zwischen Netzen unterschiedlicher Betreiber als isolierter Dienst angeboten. Die Vertrauensbeziehungen zwischen den unterschiedlichen Parteien wurden aus den Geschäftsmodellen abgeleitet. Die wesentlichen Anforderungen an die Architektur insbesondere im Hinblick auf die zu gewährleistende Sicherheit wurden beschrieben. Hierbei wurden insbesondere eine sichere beidseitige Authentifizierung, bei der nicht abstreitbare Daten für eine Abrechnung erzeugt werden, und eine Autorisierung beschrieben. Eine weitere Eigenschaft der Architektur ist die gegebene Minimierung der bei den Nutzerauthentifizierungen notwendigen Interaktionen eines Nutzers, welche dem "Single Sign On" Rechnung trägt.

Weiter wurde untersucht, inwieweit sich die Architektur mit dem Zusammenspiel von Standardlösungen und existierenden Technologien in Realität umsetzen lässt. Hierzu wurden zwei Prototypen für die sichere beidseitige Authentifizierung implementiert - zum einen zwischen einem Client und einem Server und zum anderen für die Kommunikation über „Wireless Local Area Networks“ (WLAN) und „Worldwide Interoperability for Microwave Access“ (WiMAX). Die Implementationen wurden mit Erfolg getestet.

Zum Abschluss wird ein Ausblick auf die zukünftige Entwicklung sowie weitere Anregungen gegeben.

Zusammenfassend ist zu sagen, dass auf Basis der abstrakten Lösungen mit den Prototypen gezeigt ist, wie das sichere Roaming übergangslos zwischen heterogenen Netzen aus unterschiedlichen Domänen möglich ist.

12 Danksagung

Diese Arbeit wurde in den Jahren 2002 bis 2006 am Fraunhofer Institut für Sicherheit in der Informationstechnologie SIT in Darmstadt durchgeführt. Allen Kollegen, die mir Hinweise und Ratschläge gegeben haben, sei hiermit gedankt.

Mein besonderer Dank gilt Frau Prof. Dr. C. Eckert für die Anregung zu dieser Arbeit und dafür, dass sie mir die Gelegenheit gegeben hat, sie unter anderem im Rahmen einer Forschungskooperation mit den Fujitsu Laboratories of Europe am Fraunhofer Institut für Sicherheit in der Informationstechnologie durchzuführen.

Weiter gilt mein Dank der Kollegin E. Giessler und den Kollegen M. Ilyas und Dr. M. Schneider des SIT, mit denen ich im Rahmen dieses Projektes zusammengearbeitet habe, für ihre Hinweise und Bereitschaft, diese Arbeit zu unterstützen.

13 Literaturverzeichnis

[21.133] 3GPP TS 21.133. 3G Security: Threats and Requirements

[22.022] 3GPP TS 22.022. Personalisation of Mobile Equipment (ME); Mobile functionality specification (Release 5)

[22.048] 3GPP TS 22.048. Security Mechanisms for the (U)SIM application toolkit; Stage 1 (Release 4)

[23.002] 3GPP TS 23.002. Network Architecture (Release 5)

[23.048] 3GPP TS 23.048. Security Mechanisms for the (U)SIM application toolkit; Stage 2 (Release 5)

[23.101] 3GPP TS 23.101. General UMTS Architecture(Release 4)

[23.228] 3GPP TS 23.228. IP Mulitmedia Subsystem (IMS); Stage 2 (Release 5)

[23.234] 3GPP TS 22.048: WLAN Subsystem; System Description (Release 6)

[23.934] 3GPP TS 23.934: 3GPP system to WLAN Interworking; Functional and architectural definition (Release 6), 2002

[29.002] 3GPP TS 29.002 Mobile Application Part (MAP) specification (Rel. 7)

[31.101] 3GPP TS 31.101. UICC terminal interface; Physical and logical characteristics (Release 6)

[31.111] 3GPP TS 31.111. USIM Application Toolkit (USAT) (Release 5)

[32.815] 3GPP TR 32.815 Telecommunication Management, Charging Management, Online Charging System (OCS) architecture study

[33.102] 3GPP TS 33.102. Security Architecture(Release 5)

[33.103] 3GPP TS 33.103. 3G security; Integration guidelines(Release 4)

[33.105] 3GPP TS 33.105. Cryptographic Algorithm Requirements(Release 4)

[33.200] 3GPP TS 33.200. Network domain security; MAP application layer security(Release 4)

[33.203] 3GPP TS 33.203. Access security for IP-based services (Release 5)

[33.210] 3GPP TS 33.210. Network domain security; IP network layer security (Release5)

[33.810] 3GPP TS 33.810. NDS-Authentication Framework to support IP evolution (Release 6)

[33.900] 3GPP TR 33.900. A Guide to 3rd Generation Security V1.2.0

[33.908] 3GPP TS 33.908. Design, Specification and Evaluation of 3GPP Standard Confidentiality and Integrity Algorithms

[33.cde] 3GPP TS 33.cde: Wireless Local Area Network (WLAN) Interworking Security (Release 6)

[35.201] 3GPP TS 35.201. Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 1: f8 and f9 Specification

[35.202] 3GPP TS 35.202. Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 2: KASUMI Specification

[35.203] 3GPP TS 35.203. Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 3: Implementers Test Data

[35.204] 3GPP TS 35.204. Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 4: Design Conformance Test Data

[35.205], 3GPP TS 35.205. An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 1:General

[35.206] 3GPP TS 35.206. An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 2:Algorithm Specification

[35.207] 3GPP TS 35.207. An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 3:Implementors Test Data

[35.208] 3GPP TS 35.208. An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 4:Design Conformance Test Data

[35.209] 3GPP TS 35.209. An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 5:Summary and results of Design and Evaluation

[3GPPa] Nokia, “Security solution for UTRAN IP transport”, 3GPP TSG SA WG3 Security — S3#24 S3-020359, Jul. 2002

[3GPPb] ftp://ftp.3gpp.org/specs/archive/21_series/21.133/21133-410.zip

[3GPPc] ftp://ftp.3gpp.org/specs/archive/23_series/23.057/23057-440.zip

[AboSim99] B. Aboba, D. Simon: PPP EAP TLS Authentication Protocol. IETF, RFC 2716, 1999.

[adhoc] http://www.ccs.neu.edu/home/zhufeng/security_manet.html

[AdLl99] Carlisle Adams, Steve Lloyd. Understanding public key infrastructure. New Riders Publishing, 1999

[AES] http://csrc.nist.gov/encryption/aes/

[AFMP03] James Aspnes, Joan Feigenbaum, Michael Mitzenmacher, David Parkes: Towards better definitions and measures of Internet security. Workshop

on Large-Scale-Network Security and Deployment Obstacles, March 2003

[AF02] Alvén, David; Farhang, Reza: Does it take a WISP to manage a wisp of hotspots? - Analysis of the WLAN market from a WISP perspective. Master Thesis, Department of Microelectronics and Information Technology Royal Institute of Technology, February 2002, http://www.e.kth.se/~e96_rfh/wisp_analysis.pdf, Feb. 2003

[AH03] Arkko, J., Haverinen, H.: EAP AKA Authentication. Internet-draft, 2003, http://vesuvio.ipv6.cselt.it/internet-drafts/draft-arkko-pptext-eap-aka-09.txt

[AIL+02] Adrangi, Farid; Iyer, Prakash; Leung, Kent; et al.: Problem Statement for Mobile IPv4 Traversal Akreuz VPN Gateways. Internet Draft < draft-ietf-mobileip-vpn-problem-statement-00>, March 2002

[Anand01] Nikhil Anand, An Overview of Bluetooth Security, February 22, 2001; www.warchalking.com.br/tutingles/pdf/Nikhil_Anand_GSEC.pdf

[Ande94a] Ross J. Anderson: Why cryptosystems fail. Communications of the ACM, Vol. 37, Issue 11, November 1994

[Ande94b] Ross Anderson: “A5 (was hacking digital phones)”, sci.crypt, 17. Juni 1994

[Ande01] Ross Anderson: Why Information Security is Hard — An Economic Perspective. 17th Annual Computer Security Applications Conference (ACSAC 2001), Proceedings, December 2001

[ANSA05] Jari Arkko, Pasi Eronen, Rainer Falk Michael Georgiades, Seppo Heikkinen Elisa, Ian Herwono, Günther Horn. Keith Howker, Mattias Johansson, Rieks Joosten, Geert Kleinhuis, Florian Kohlmayer, Mika Kousa, Julien Laganier, Tim Leinmüller, Daniel Migault, Anand Prasad, Mark Priestley, Peter Schoo, Göran Selander, Kristian Slavov, Hannes Tschofenig, Tseno Tsenov, Alf Zugenmaier: Sixth Framework Program, Priority IST-2002-2.3.1.4, Mobile and Wireless Systems beyond 3G, Project 507134, WWI Ambient Networks, Deliverable 7.2 Ambient Network Security Architecture, Annex 2 Security Requirements, Concepts and Solutions for Secure Access and Mobility Procedures, 29.12.2005

[ANSA06] Jari Arkko, Pasi Eronen, Rainer Falk Michael Georgiades, Seppo Heikkinen Elisa, Ian Herwono, Günther Horn. Keith Howker, Mattias Johansson, Rieks Joosten, Geert Kleinhuis, Florian Kohlmayer, Mika Kousa, Julien Laganier, Tim Leinmüller, Daniel Migault, Anand Prasad, Mark Priestley, Peter Schoo, Göran Selander, Kristian Slavov, Hannes Tschofenig, Tseno Tsenov, Alf Zugenmaier:Sixth Framework Program, Priority IST-2002-2.3.1.4, Mobile and Wireless Systems beyond 3G,

Project 507134, WWI Ambient Networks, Deliverable 7.2 Ambient Network Security Architecture, Final Version, 08.02.2006

[APPNEL] Timothy Appnel: “Introducing MIDP 2.0” 18/12/2002 http://www.onjava.com/pub/a/onjava/2002/12/18/midp.html

[Araya95] Agustin A. Araya: Questioning ubiquitous computing. 23rd ACM Annual Conference on Computer Science, Proceedings, ACM Press, 1995

[Arazi77] Arazi, B. : Handwriting Identif. By Means of Run-Length Measurements, IEEE Trans. on Syst. Man, and Cybernetics, Vol.7 (1977)

[ArbSha01] William A. Arbaugh, Narendar Shankar, Y.C. Justin Wan, “Your 802.11 Wireless Network has No Clothes”, March, 2001

[ASN02] Abid M.; Sulistyo S.; Najib W.: UMTS Security, Security in Core Network and UTRAN, November 2002

[AWM01] Abdullah N. Alghannam, Michael E. Woodward, J. E. Mellor: Security as a QoS Routing Issue. PG Net 2001, June 2001

[BanBer02] Guruduth Banavar, Abraham Bernstein: Software infrastructure and design challenges for ubiquitous computing applications. Communications of the ACM, Volume 45, No. 12, December 2002

[BarDey03] Louise Barkuus, Anind Dey: Location-based Services for Mobile Telephony: a Study of Users' Privacy Concerns. 9th IFIP TC13 International Conference on Human-Computer Interaction, INTERACT 2003, July 2003

[Beinat01] Euro Beinat: Privacy and Location-based Services – Stating the Policies clearly. GEO Informatics, Vol. 4, Issue 6, September 2001

[BelRog93] Mihir Bellare, Phillip Rogaway: Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. Proceedings of the 1st ACM conference on computer and communications security (CCS '93)", ACM Press, November 1993

[BGW99] Briceno, Marc, Ian Goldberg, David Wagner: A pedacogical Implementation of the GSM A5/1 and A5/2 voice privacy encryption algorithms, http://www.mirrors.wiretapped.net

[BHL03] Russell R. Barton, William J. Hery, Peng Liu: An S-vector for web application security management. ACM Workshop on Business Driven Security Engineering (BIZSEC 2003), Proceedings, ACM Press, 2003

[BHV01] Deborah Bodeau, Ronda Henning, Rayford Vaughn: Workshop on Information Security System Scoring and Ranking -- Proceedings. Applied Computer Security Associates (ACSA), 2001

[BEGH+04] Bayarou, Kpatcha M. / Enzmann, Matthias / Giessler, Elisabeth / Haisch, Michael K.D. / Hunter, Brian / Ilyas, Mohammad / Rohr, Sebastian /

Schneider, Markus: Towards Certificate-based Authentication for Future Mobile Communications Wireless Personal Communications, Special Issue on Security for Next Generation Communications, Kluwer, 2004

[BEPR+03] Bayarou, Kpatcha M. / Eckert, Claudia / Prasad, Anand / Rohr, Sebastian / Schoo, Peter / Wang, Hu: Feasible and Meaningful Combinations of Access and Network Technologies for Future Mobile Communications Conference WWRF 10: Co-operative and Ad-Hoc Networks, New York Oct 2003

[BEPR+04] Bayarou, Kpatcha M. / Eckert, Claudia / Prasad, Anand / Rohr, Sebastian / Schoo, Peter / Wang, Hu: 3D and WLAN Interworking: Towards a Secure Solution for Tight Coupling The 7th International Symposium on Wireless Personal Multimedia Communications WPMC 2004, Abano Terme, Italy 12-15 Sep 2004

[BD02] Braun, T., Danzeisen, M., Secure Mobile IP Communication, http://anaisoft.unige.ch/public-documents/deliverables , 2002

[BDRS+96] Matt Blaze, Whitfield Diffie, Ronald L. Rivest, Bruce Schneier, Tsutomu Shimomura, Eric Thompson, Michael Wiener: Minimal key lengths for symmetric ciphers to provide adequate commercial security. http:// www.bsa.org / policy / encryption / cryptographers_c.html, January 1996

[BEL02] S. M. Bellovin, et al., “On the Use of SCTP with IPsec”, Internet draft, work in progress, draft-ietf-ipsec-sctp-04.txt (expires in Apr. 2003)

[Bel96] Bellovin, S.: Problem Areas for the IP Security Protocols, USENIX UNIX Security Symposium, San Jose, California, July 1996.

[BeLM01] Diana Berbecaru, Antonio Lioy, Marius Marian. On the Complexity of Public-Key Certificate Validation. Information Security (ISC01), 4th International Conference, LNCS 2200, Springer Verlag, 2001

[BenOFe02] Paul Bender and Stephen O’Fee: European Regulation of Software Radio, In “Software Defined Radio: Origins, Drivers and International Perspectives”, Edited by Walter Tuttlebee, John Wiley & Sons, LTD, 2002.

[Bey00a] Bey, C., K. Dupre: Palm File Format Specification, Palm Inc. 5400 Bayfront Plaza, Santa Clara CA 95052, Dokument Nr. 3008-003, Mai 2000

[Bey00b] Bey, C., E. Freeman, J. Ostrern: Palm OS Programmers Companion, Palm Inc. 5400 Bayfront Plaza, Santa Clara CA 95052, Dokument Nr. 3004-003, Juni 2000

[Bey00c] Bey, C., E. Freeman, J. Ostrern: Palm OS Reference, Palm Inc. 5400 Bayfront Plaza, Santa Clara CA 95052, Dokument Nr. 3003-003, Juni 2000

[BHH+02] Boman K.; Horn G.; Howard P.; Niemi V.:UMTS security, Electronics & Vommunication Engineering Journal, October 2002
[BemTeuPlaPeePed02] Jeroen van Bemmel, Harold Teunissen, Dirk-Jaap Plas, Bastien Peelen, Arjan Peddemors: A Reference Architecture for 4G Services. Wireless World Research Forum (WWRF 7), December 2002.

[Bja2001] Geir Stian Bjåen, Erling Kaasin, Security in GPRS, Master Thesis, May 2001

[Bluetooth] http://www.bluetooth.org/spec/

[Blue01] Muller, Nathan J: Bluetooth, MITP-Verlag GmbH, Bonn, 2001

[BL03] Business Layers: Provisioning: The best foundation for password management. White Paper, 2003

[BNH+03] S. Blake-Wilson, M. Nystrom, D. Hopwood, J. Mikkelsen, T. Wright: Transport Layer Security (TLS) Extensions. RFC 3546, June 2003

[Boneh98] Dan Boneh: The decision Diffie-Hellman problem. Proceedings of the Third Algorithmic Number Theory Symposium, LNCS 1423, Springer Verlag, 1998

[BruBon03] David Brumley, Dan Boneh: Remote Timing Attacks Are Practical. 12th USENIX Security Symposium 2003, Proceedings, August 2003

[BSI06] Bundesamt für Sicherheit in der Informationstechnik (BSI): Drahtlose Kommunikationssysteme und ihre Sicherheitsaspekte, 2006

[BSSW03] Stefan Berger, Henning Schulzrinne, Stylianos Sidiroglou, Xiaotao Wu: Ubiquitous Computing Using SIP. 13th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'03), Proceedings, ACM Press, June 2003

[BLM01] Diana Berbecaru, Antonio Lioy, Marius Marian: On the Complexity of Public-Key Certificate Validation. Information Security (ISC01), 4th International Conference, Proceedings, LNCS 2200, Springer Verlag, 2001.

[BNHM03] S. Blake-Wilson, M. Nystrom, D. Hopwood, J. Mikkelsen, T. Wright. Transport Layer Security (TLS) Extensions. RFC 3546, June 2003

[Bonser2002] Wayne Bonser, US Defense Initiatives in Software Radio(SDR), In “Software Defined Radio: Origins, Drivers and International Perspectives”, Edited by Walter Tuttlebee, John Wiley & Sons, LTD, 2002.

[BRAIN22] A. Lopez, et al., “BRAIN architecture specifications and models, BRAIN functionality and protocol specification”, IST-1999-10050 project BRAIN deliverable D2.2, Mar. 2001, available at

http://jungla.dit.upm.es/~ist-brain/deliverables/BRAIN%20Del%202.2.pdf

[HiLAN] http://portal.etsi.org/radio/HiperLAN/HiperLAN.asp

[Brands00] Stefan Brands: Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy. MIT Press, August 2000

[Brazier99] John R.T. Brazier: Possible NSA decryption capabilities. http:// jya.com / nsa-study.htm, June 1999

[Bre02] Brewin, B.: New Wi-Fi security would do little for public 'hot spots'. http://www.computerworld.com/securitytopics/security/story/0,10801,75535,00.html, Oct. 2002

[BRENNAN] S. Brennan : “Intel Faces Silicon Limitations”

[Brookson01] Charles Brookson, GPRS Security, Dec. 2001

[Bruyn85] Bruyne, P. de, Signature Verification Using Holistic Measures, Computers and Security, Vol.4 (1985)

[BSI00] Bundesamt für Sicherheit und Informationstechnik, Technische Evaluierungskriterien zur Bewertung und Klassifizierung biometrischer Systeme, Bonn, 2000

[BSI03] Andreas Schmidt, Bundesamt für Sicherheit und Informationstechnik, Zertifizierungsinfrastruktur für die PKI-1-Verwaltung, Bonn, 2003

[BT-WP2002] Bluetooth SIG Security Expert Group, “Bluetooth™ Security White Paper”, April 2002

[BTH03] Jeroen van Bemmel, Harold Teunissen, Gerard Hoekstra: Security aspects of 4G services. Wireless World Research Forum (WWRF #9), July 2003

[BTPP+02] Jeroen van Bemmel, Harold Teunissen, Dirk-Jaap Plas, Bastien Peelen, Arjan Peddemors: A reference architecture for 4G services. Wireless World Research Forum (WWRF #7), Dezember 2002

[Buchm99] Buchmann, J., Einführung in die Kryptographie, Springer Verlag, Berlin, 1999

[Bur99] Burge, M. , W. Burger : Ear Biometrics in: Jain, A., R. Bolle,S. Pankanti: Biometrics: Personal Identification in Networked Society, Kluywer Academic Publishers, 1999

[Butler02] Shawn A. Butler: Security Attribute Evaluation Method: A Cost-Benefit Approach. Proceedings of the 24th international conference on Software engineering (ICSE), ACM Press, 2002

[BWBD04] Nilanjan Banerjee, Wei Wu, Kalyan Basu, Sajal K. Das: Analysis of SIP-based mobility management in 4G wireless networks. Computer Communications, Vol. 27, No. 8, May, 2004 [BER04] Bayarou,

Kpatcha M. / Eckert, Claudia / Rohr, Sebastian: NGN, AII-IP, B3G: Enabler für das Future Net?! Überblick über Entwicklungen im Bereich zukünftiger Netze Informatik-Spektrum, Band 27, Heft 1, Springer Verlag, Heidelberg Feb 2004

[BWIF_Spec] http://www.bwif.org/

[CAG+02] Calhoun, P.; Arkko, J.; Guttmann, E.; Zorn, G.; Lughney, J.: Diameter Base Protocol, Juni 2002

[Cam00] A. Campbell, J. Gomez, C. Y. Wan, S. Kim, Z. Turanyi, and A. Valko, "Cellular IP", Internet draft, Jan. 2000, <expired>, available at http://comet.columbia.edu/cellularip/pub/draft-ietf-mobileip-cellularip-00.txt

[Cam01] A. T. Campbell, and J. Gomez, "IP Micro-Mobility Protocols", ACM SIGMOBILE Mobile Computer and Communication Review (MC2R), 2001, http://comet.columbia.edu/micromobility/pub/survey.pdf

[Cam02] A. T. Campbell, J. Gomez, S. Kim, C. Y. Wan, Z. R. Turanyi, and A. G. Valko, "Comparison of IP Micro-Mobility Protocols", IEEE Wireless Communications Magazine, Vol. 9, No. 1, Feb. 2002

[CamHer02] Jan Camenisch, Els van Herreweghen: Design and Implementation of the idemix anonymous credential system. ACM CCS'02, ACM Press, 2002

[CamLys01] Jan Camenisch, Anna Lysyanskaya: An efficient system for non-transferable anonymous credentials with optional anonymity revocation. Advances in Cryptology — EUROCRYPT 2001, LNCS 2045, Springer Verlag, 2001

[CamLys02] Jan Camenisch, Anna Lysyanskaya: Dynamic accumulators and application to efficient revocation and anonymous credentials. Advances in Cryptology — CRYPTO 2002, LNCS 2442, Springer Verlag, 2002

[CanKem03a] Scott Cantor, John Kemp: Liberty ID-FF Bindings and Profiles Specification, Version 1.2. www.projectliberty.org, 2003

[CanKem03b] Scott Cantor, John Kemp: Liberty ID-FF Protocols and Schema Specification, Version 1.2. www.projectliberty.org, 2003

[Cas98] C. Castelluccia, "A Hierarchical Mobile IPv6 Proposal", INRIA Rapport Techinique, No.0226, Nov. 1998

[Carter00] James W. Carter, "The Security of IEEE 802.11 and Bluetooth", Nov. 2000

[Cas00] C. Castelluccia, "HMIPv6: A Hierarchical Mobile IPv6 Proposal", Feb. 2000

[CGH98] Ran Canetti, Oded Goldreich, Shai Halevi: The random oracle methodology, revisited. 30th Annual ACM Symposium on Theory of Computing (STOC'98), ACM Press, 1998

[Chandrasiri02] Pubudu Chandrasiri: First Steps to Software Defined Radio Standards: MExE, the Mobile Execution Environment, In "Software Defined Radio: Origins, Drivers and International Perspectives", Edited by Walter Tuttlebee, John Wiley & Sons, LTD, 2002.

[ChaEve86] David Chaum, Jan-Hendrik Evertse: A secure and privacy-protecting protocol for transmitting personal information between organizations. Advances in Cryptology — CRYPTO'86, LNCS 263, Springer Verlag, 1986

[Chaum89] David Chaum: Privacy Protected Payments: Unconditional Payer and/or Payee Untraceability. Smart Card 2000, Proceedings, North Holland, 1989

[Che95] Chen, Yi-an: A Survey Paper on Mobile IP, 1995 http://www.cs.wustl.edu/~jain/cis788-95/mobile_ip/index.html

[ChOt02] David W. Chadwick, Alexander Otenko. The PERMIS X.509 Role Based Privilege Management Infrastructure. SACMAT'02, Proceedings, ACM Press, June, 2003

[CJP02] Calhoun, Pat R.; Johansson, Tony; Perkins, Charles E.: Diameter Mobile IPv4 Application. Internet Draft <draft-ietf-aaa-diameter-mobileip-13.txt>, October 2002

[Clercq02] Jan de Clercq: Single Sign-On Architectures. Infrastructure Security Conference (InfraSec 2002), Proceedings, LNCS 2437, Springer Verlag, 2002.

[CLGZ02] Pat R. Calhoun, John Loughney, Erik Guttman, Glen Zorn, Jari Arkko. Diameter Base Protocol. AAA Working Group, Internet-Draft, draft-ietf-aaa-diameter-17.txt, (www.ietf.org), December, 2002

[CMR04] Huseyin Cavusoglu, Birendra Mishra, Srinivasan Raghunathan: A Model for Evaluating IT Security Investments. Communications of the ACM, Vol. 47, No. 7, July 2004

[Com91] D. E. Comer, "Internetworking with TCP/IP, Volume I; Principles, Protocols, And Architecture", Second Edition, Prentice Hall International Editions, 1991

[CRHB03] Andy Crabtree, Tom Rodden, Terry Hemmings, Steve Benford: Finding a place for UbiComp in the home. 5th International Conference on Ubiquitous Computing (UbiComp 2003), Proceedings, LNCS 2864, Springer Verlag, 2003

[CZ06] Mobility braucht eine Gesamtstrategie, Computerzeitung Nr 29 , 17.7.2006

[CZP+01] Calhoun, P.; Zorn, G.; Pan, P.; Akthar, H.: Diameter Framework Document, März 2001

[Dav01] Davis C.: IPsec: Securing VPNs. McGraw-Hill, 2001

[Dav02] Davoli, Mario: WLAN as a Complement to GPRS and 3G Services. White Paper, 2002, http://broadcastpapers.com/whitepapers/WLAN-as-a-Complement-to-GPRS-and-3G-Services-White-Paper.cfm?objid=32&pid=130&fromCategory=25

[DaPr02] B.A. Davey, H.A. Priestley. Introduction to Lattices and Order. Cambridge University Press, 2002

[DavPri02] B.A. Davey, H.A. Priestley: Introduction of lattices and order. Cambridge University Press, 2002

[DavRav00] Nigel Davies, Pierre-Guilaume Raverdy: Position Paper: The role of Platforms and Operating Systems in Supporting Home Networks, Proceedings of the 9th ACM SIGOPS European workshop: Beyond the PC: new challenges for the operating system, September 2000

[DCY04] Ram Dantu, João Cangussu, Arun Yelimeli: Dynamic Control of Worm Propagation. International Conference on Information Technology: Coding and Computing (ITCC'04), Proceedings, 2004

[DECT] http://portal.etsi.org/radio/DECT/dect.asp

[DH00] Doraswamy, N.; Harkins D.: IPsec: Der neue Sicherheitsstandard für das Internet, Intranet und virtuelle private Netzte. Addison-Wesley, 2000

[DH76] Diffie, W., Hellman, M.: New Directions in Cryptography. IEEE Transactions on Information Theory, Vol. 22, pp.644-654, November 1976.

[DifHel76] Whitfield Diffie, Martin Hellman: New Directions in Cryptography. IEEE Transactions of Information Theory, Vol. 22, No. 6, November 1976

[DieAll99] T. Dierks, C. Allen: The TLS Protocol. Version 1.0 IETF RFC 2246, January 1999

[DilBou2002] Markus Dillinger and Didier Bourse: Reconfigurable Radio in Europe, In "Software Defined Radio: Origins, Drivers and International Perspectives", Edited by Walter Tuttlebee, John Wiley & Sons, LTD, 2002.

[Dixit02] S. Dixit und R. Prasad, Wireless IP and Building the Mobile Internet, Artech House, Boston, USA, 2002.

[DNSSec] D. Eastlake, "Domain Name System Security Extensions," RFC 2535, Mar. 1999.

[DNSSEC1] http://www.dnssec.net/

[DoD85] Department of Defense: Department of Defense Trusted Computer System Evaluation Criteria. Department of Defense Standard, DoD 5200.28-STD, December 1985

[DolYao83] Danny Dolev, Andrew C. Yao: On the security of public key protocols. IEEE Transactions on Information Theory, Vol. 29, No. 2, March 1983

[DeKu1996] Marc Delprat; Vinod Kumar: Second Generation Systems in The mobile Communications Handbook Ed. Jerry D. Gibson, 1996, CRC Press Inc.

[Eckert04] C. Eckert, IT-Sicherheit: Konzepte, Verfahren, Protokolle Oldenbourg Wissenschaftsverlag, München

[Eckert01] Eckert, C. : Zur Sicherheit mobiler persönlicher Endgeräte – eine Bestandsaufnahme, in: Horster, P. : Kommunikationssicherheit im Zeichen des Internets, Vieweg Verlag, März 2001, ISBN 3-528-05763-7

[Eckert02] Eckert, C. : Vorlesung zur IT-Sicherheit, Sommersemester 2002

[Eco03] eco: AK WLAN Zielsetzung. www.eco.de/servlet/PB/menu/1110486_l1/index.html

[EFLR99] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, T. Ylonen. SPKI Certificate Theory. RFC 2693, September, 1999

[EITO03] European Information Technology Observatory: European Information Technology Observatory 2003.

[Eklund] C. Eklund et. al., “IEEE Standard 802.16: A Technical Overview of the WirelessMAN™ Air Interface for Broadband Wireless Access,” IEEE Communications Magazine, June 2002, pp. 98-107.

[Ellis00] Ellison, C., B. Schneier : Ten Risks of PKI, Comp Security J., Vol.16, Nr.1 (2000)

[Ellison99] C. Ellison. SPKI Requirements. RFC 2692, September, 1999

[Ellison03] Gary Ellison: Liberty ID-WSF Security Mechanism, Version 1.0. www.projectliberty.org, 2003

[EGHH+04] Enzmann, Matthias / Giessler, Elisabeth / Haisch, Michael K.D. / Hunter, Brian / Ilyas, Mohammad / Schneider, Markus: A Note on Certificate Path Verification in Next Generation Mobile Communications, 17th Intenational Conference on Architecture of Computing Systems (ARCS), 23.-26. March, Augsburg, Germany, Proceedings, Springer Verlag, LNCS 2981, Mar 2004

[EKMP+03a] Kevin Eustice, Leonard Kleinrock, Shane Markstrum, Gerald Popek, V. Ramakrishana, Peter Reiher: Securing Nomads: The Case for Quarantine, Examination, and Decontamination. New Security Paradigms Workshop 2003, Proceedings, 2003

[EKMP+03b] Kevin Eustice, Leonard Kleinrock, Shane Markstrum, Gerald Popek, Venkatraman Ramakrishna, Peter Reiher: Enabling Secure Ubiquitous Interactions. 1st International Workshop on Middleware for Pervasive and Ad-Hoc Computing, Proceedings, July 2003

[ElGamal85] Taher ElGamal: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions of Information Theory, Vol. 31, No. 4, July 1985

[Eriksson2000] H. Eriksson. 3G Services and the Roadmap Ahead. In Visions of the Wireless World, Brussels, Belgium, Workshop of the Wireless Strategic Initiative, Dec. 12, 2000

[Eri01] "IPv6 in 3G Wireless Networks", White Paper, available at http://www.ipv6tf.org/

[Esc01] Escudero A.: Location Privacy in IPv6: Tracking binding updates. IDMS2001. Lancaster. UK. September 2001

[ETSI98] General Packet Radio Service (GPRS); Service Description; Stage 2, European Telecommunications Standards Institute, TS 101 350, August 1998

[ETSI-DAB] ETS 300 401(May 1997) Radio broadcasting systems; Digital Audio Broadcasting (DAB) to mobile, portable and fixed receivers; http://www.etsi.org/

[ETSI-DVB] ETSI EN 300 744 V1.4.1 (2001-01) : Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television; http://www.etsi.org/

[Eurescom_P912] Eurescom Project P912-PF, Security for mobility in IP

[EURES-LUPA2001] EURESCOM Strategic Study P1046-LUPA Local provision of 3G and 3G+

[FanGer02] Andrew Fano, Anatole Gershman: The future of business services in the age of ubiquitous computing. Communications of the ACM, Volume 45, No. 12, December 2002

[FCC2001SDR] US-FCC (Federal Communication Commission) – 1st. Report and Order In the matter of Authorization and Use of Software Defined Radios http://www.fcc.gov/Bureaus/Engineering_Technology/Orders/2001/

[Fed06] Digitale Signaturen und Public Key Infrastrukturen, Sicherheitsmanagement und IT Sicherheit, Vorlesung von Prof. Dr.-Ing. Hannes Federrath, Universität Regensburg

[FeGiLy92] Ferraiolo, D., D, Gilbert, and N. Lynch: An Examination of Federal and Commercial Access Control Policy Needs, in Proceedings of the NIST-NSA National (USA) Computer Security Conference 1992

[FeKuCh03] Ferraiolo, David, Richard Kuhn, Ramaswamy Chandramouli: Role Based Access Control, Artech House, Boston, 2003

[Fer02] B. E. Fernandes, "Mobile Wireless Working Group", Tutorial Workshop on IPv6, Jan. 2002, Geneva, Switzerland, available at

http://www.ec.ipv6tf.org/PublicDocuments/IPv6_Mobile_Wireless_WG.pdf

[FIPS-199] National Institute of Standards and Technology (NIST), Computer Security Division, Information Technology Laboratory: Standards for Security Categorization of Federal Information and Information Systems. FIPS PUB 199, Federal Information Processing Standards Publication, December 2003

[FLL03] E. Giessler, M. Haisch, M. Ilyas, M. Schneider: “Seamless Security Roaming“, Deliverable D4, SIT, Darmstadt 2003, unveröffentlicht

[FOMA1] FOMA Technology http://www.nttdocomo.com/technologies/present/fomatechnology/index.html

[FOR99] Ford, M.: Securing a Mobile Internet. 1999 http://www.eurescom.de/~public-webspace/P900-series/P912/Papers

[FSGK+01] David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, D. Richard Kuhn, Ramaswamy Chandramouli: Proposed NIST standard for role-based access control. ACM Transactions on Information and System Security, Vol. 4, No. 3, ACM Press, August, 2001.

[FTD03] Matthias Deiß: Inside Business – Lieber geklaut als schlecht gemacht. Financial Times Deutschland, 4th of October, 2003

[FunCheSir02] Y.F. Fung, W.L. Cheung, H.R. Sirisena: A study of IEEE 1394 for network computing. Power Systems and Communications Infrastructures for the Future, Beijing, September 2002

[Furukawa2000] Makoto Furukawa: Expectation to All IP NW Expectation to All IP NW and its evolution and its evolution http://www.3gpp.org/ftp/workshop/Archive/0002IP/Docs/PDF/AIP-000032.pdf

[FVCG00] S. Farrell, J. Vollbrecht, P. Calhoun, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, D. Spence: AAA Authorization Requirements, RFC 2906, August 2000

[GaSi96] Eran Gabber, Abraham Silberschatz. Agora: A Minimal Distributed Protocol for Electronic Commerce. In Proceedings of the Second USENIX Workshop on Electronic Commerce, November 1996.

[Gei02a] Geier, J.: 802.11 Security Beyond WEP. http://metatag.tripod.com/howto/howto_802_1_1_Security_Beyond_WEP.htm, June 2002

[Gei02b] Geier, J.: Minimizing WLAN Security Threats. http://www.80211-planet.com/tutorials/article.php/1457211, September 2002

[Gellman02] Robert Gellman: Privacy, Consumers, and Cost -- How the Lack of Privacy Costs Consumers and Why Business Studies of Privacy Costs are Biased and Incomplete, March 2002 http://www.epic.org/reports/dmfprivacy.pdf

[GHSZ+04] Manfred Glesner, Thomas Hollstein, Leandro Soares Indrusiak, Peter Zipf, Thilo Pionteck, Mihail Petrov, Heiko Zimmer, Tudor Murgan: Reconfigurable platforms for ubiquitous computing. 1st Conference on Computing Frontiers (CF'04), Proceedings", ACM Press, April 2004

[Gho02] Ghosh, D.: Mobile IP, ACM Crossroads, 2002, http://www.acm.org/crossroads/xrds7-2/mobileip.html

[Gla2000] S. Glass, et al, Mobile IP Authentication, Authorization, and Accounting Requirements, RFC2977, Oct. 2000

[Goli97] Golic, Jovan Dj., Cryptoanalysis of Alleged A5 Stream Cipher, Proceedings of EUROCRYPT'97, LNCS 1233, S.239-255, Springer Verlag, 1997

[GorLoe02] Lawrence A. Gordon, Martin P. Loeb: The economics of information security investment. ACM Transactions on Information and System Security, Vol. 5, No. 4, November 2002

[Gos02] Goswami, Subrata: DIAMETER Application for Mobile-IPv4 and 802.11 Authentication. Internet Draft <draft-goswami-aaa-mipv4-wlan-auth-01.txt>, Oct. 2002

[Grable2002] Mike Grable, Regulation of Software Defined Radio – United States, In "Software Defined Radio: Origins, Drivers and International Perspectives", Edited by Walter Tuttlebee, John Wiley & Sons, LTD, 2002

[GRPuGu02] Kai J. Grahn; Göran Pulkkis, Jean Sebastien Guillard, Security of Mobile and Wireless Networks, Juni 2002 http://proceedings.informingscience.org/IS2002Proceedings/papers/Grahn152Secur.pdf

[Grilo] A. Grilo et.al., "terminal Independent Mobility for IP (TIMIP)," IEEE Comm. Mag., Dec. 2001, pp. 34-41.

[GSM01a] GSM 03.20 Security related network functions, ETSI TS 100 929 V8.1.0, Jul.2001

[GSM01b] GSM 02.09 Security aspects, ETSI TS 100 920 V8.0.1, Jun.2001

[GSM03] Timothy Grance, Marc Stevens, Marissa Myers: Guide to Selecting Information Technology Security Products — Recommendations of the National Institute of Standards and Technology. NIST Special Publication 800-36, October 2003

[Gura04] Nils Gura, Arun Patel, Arvinderpal Wander, Hans Eberle, Sheueling Chang Shantz: Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs in LNCS 3156, Springer Verlag Berlin, 2004

[Haisch01] Konzeption und Entwicklung von Verfahren zur Absicherung handschriftlicher Merkmale bei der Benutzerauthentifikation auf mobilen Endgeräten, Diplomarbeit TU-Darmstadt, August 2001

[HaiSteiVie02] C. Vielhauer, M.Haisch, R. Steinmetz: Schlüsselmanagement in biometrischen Systemen Paderborn (Enterprise Security 2002) Mrz. 2002

[HalCer04] R.S. Hall, H. Cervantes: An OSGi Implementation and Experience Report. Proceedings of IEEE Consumer Communications and Networking Conference, January 2004

[HAVi99] HAVi the A/V digital network revolution -- Technical background. Whitepaper, 1999, retrieved from WWW: http://www.havi.org/pdf/white.pdf

[Hansen00] Hansén, H.: IPsec and Mobile-IP in Mobile Ad Hoc Networking, 2000

[HasJähZanSti01] Hasan, Jürgen Jähnert, Sebastian Zander, Burkhard Stiller: Authentication, Authorization, Accounting, and Charging for the Mobile Internet. IST Mobile Communications Summit, Proceedings, September, 2001.

[HAWAII] R. Ramjee, T. La Porta, S. Thuel, K. Varadhan, and L. Salgarelli, “IP micro-mobility support using HAWAII”, Internet draft, Jul. 2000, <expired>, http://www.ietf.org/proceedings/00jul/I-D/mobileip-hawaii-01.txt

[HC98] Harkins, D.; Carrel, D.: The Internet Key Exchange (IKE). Request for Comments 2408, November 1998

[Henning99] Ronda R. Henning: Security Service Level Agreements: Quantifiable Security for the Enterprise. New Security Paradigms Workshop 1999, ACM Press, September 1999

[HHS] Hotspot Hamburg: http://www.hamburg-hotspot.de

[Hil01] Hill, J.: An Analysis of the RADIUS Authentication Protocol.2001. http://www.untruth.org/~josh/security/radius/radius-auth.html

[Hiller2001] T. Hiller, et al, Cdma2000 Wireless Data Requirements for AAA, RFC3141, Jun. 2001.

[HiperLANi] http://home.no.net/coverage/Hiperlan2%20introduction.htm

[HJZS01] Hasan, Jürgen Jähnert, Sebastian Zander, Burkhard Stiller. Authentication, Authorization, Accounting, and Charging for the Mobile Internet. IST Mobile Communications Summit, Proceedings, September, 2001

[HMIP] H. Soliman, C. Castelluccia, K. El-Malki, and L. Bellier, “Hierarchical Mobile IPv6 mobility management (HMIPv6)”, Internet draft, Oct., 2002, <work in progress>, http://www.potaroo.net/ietf/all-ids/draft-ietf-mobileip-hmipv6-07.txt

[HonLan04] Jason I. Hong, James A. Landay: An Architecture for Privacy-Sensitive Ubiquitous Computing. Second International Conference on Mobile Systems, Applications, and Services (MobiSys'04), ACM Press, June 2004

[HouArb03] Russ Housley, William Arbaugh: Security Problems in 802.11-based Networks. Communications of the ACM, Vol. 46, No. 5, May 2003

[HouPolForSol02] R. Housley, W. Polk, W. Ford, D. Solo: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. IETF, RFC 3280, 2002.

[HPFS02] R. Housley, W. Polk, W. Ford, D. Solo. Internet X.509 Public Key Infrastructure — Certificate and Certificate Revocation List (CRL) Profile, RFC 3280, April 2002

[HS02] Haverinen, H.; Salowey, J.: EAP SIM Authentication. Internet Draft: draft-haverinen-pppext-eap-sim-08.txt, Dec. 2002

[HSDPA] http://www.umtsworld.com/technology/hsdpa.htm

[HU02] Henry, Paul S.; Luo, Hui: WiFi: What’s Next?. IEEE Communications Magazine, Dec. 2002

[Huh01] Huhtanen, Karri: Security problems and solutions in WLAN access zones. http://erwin.ton.tut.fi/kh/interests/security/security-problems-and-solutions-in-wlan-access-zones.pdf, May 2001

[HuFi02] Brian Hunter, Bartol Filipovic. Enabling PKI Services for Thin-Clients. In Datenschutz und Datensicherheit (DuD), 26(9), September, 2002

[Hunt02] Brian Hunter. Simplifying PKI Usage through a Client Server Architecture and Dynamic Propagation of Certificate Paths and Repository Addresses. Trust and Privacy in Digital Business (TrustBus 2002), 13th International DEXA Workshop, IEEE Computer Society Press, September, 2002

[HZ02] Hiller, T.; Zorn, G.: Diameter Extensible Authentication Protocol (EAP) Application, Juni 2002

[Inf02] Infocomm Development Authority of Singapore: Mobile Wireless 2002-2007. An Infocomm Technology Roadmap Report, Release November 2002

[Int02] Interlink: Introduction to Diameter. 2002 http://www.interlinknetworks.com/resource/wp5-1-1.htm

[NaDoHa00] Nagannand, Doraswamy, Dan Harkins, IPSec, Addison-Wesley, München, 2000

[IPv6b4] Connection of IPv6 Domains via IPv4 Clouds, RFC 3056.

[IrDA] http://www.irda.org

[IrvLev00] Cynthia Irvine, Timothy Levin: Quality of Security Service. New Security Paradigms Workshop 2000, Proceedings, ACM Press, 2001

[IST_Arrows] European IST project ARROWS: http://www.arrows-ist.upc.es/ Deliverable 1, Security requirements for the introduction of mobility to IP, http://www.eurescom.de/~pub-deliverables/P900-series/P912/D1/p912d1.pdf

[IST-SCOUT] European IST project , http://www.ist-scout.org/

[IAPP] IEEE P802.11f, Draft Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Akreuz Distribution Systems Supporting IEEE 802.11 Operation, January 2003.

[ITU02] The Wireless World Research Forum (WWRF), ITU Seminar, Ottawa, May 28, 2002

[ITUT03] ITU-T Study Group 12: ITU-T Recommendation G.114, International Telecommunication Union, 2003

[ITUT97] ITU-T Recommendation X.509. Information Technology Open Systems Interconnection The Directory: Authentication Framework. June, 1997

[itute] http://www.itu.int/osg/spu/ni/3G/technology/index.html

[JaOd97] Stanislaw Jarecki, Andrew Odlyzko. An efficient micropayment system based on probabilistic polling. In Financial Cryptography, First International Conference (FC '97), Proceedings, number 1318 in LNCS. Springer Verlag, 1997

[JC97] Jacobs, S., Cirincione, G., Security of current Mobile IP solutions. Proc. of MILCOM'97, Vol.3, pp. 1122-1128, 1997

[JelWil98] George F. Jelen, Jeffrey R. Williams: A practical approach to measuring assurance. 14th Annual Computer Security Applications Conference (ACSAC'98), Proceedings, December 1998

[JesRob02] Leonard M. Jessup, Daniel Robey: The relevance of social issues in ubiquitous computing environments. Communications of the ACM, Volume 45, No. 12, December 2002

[Jon02] Jones, Ste: Flaws within the Dynamic Host Configuration Protocol, 2002, http://www.networkpenetration.com/dhcp_flaws.html

[JP95] Johnson, D. , Perkins, C.: Route Optimisation in Mobile IP. Internet Draft, July 1995.

[JPW02] C. Jennings, J. Peterson, M. Watson: Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks. IETF, RFC 3325, November 2002

[JRS03] Ari Juels, Ronald L. Rivest, Michael Szydlo: The Blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy. 10th ACM conference on Computer and communication security (ACM CCS'03), Proceedings, ACM Press, 2003

[KA98a] Kent, S.; Atkinson R.: Security Architecture for the Internet Protocol. Request for Comments 2401, November 1998

[KA98b] Kent, S.; Atkinson R.: IP Authentication Header. Request for Comments 2402, November 1998

[KA98c] Kent, S.; Atkinson R.: IP Encapsulating Security Payload (ESP). Request for Comments 2406, November 1998

[KAG98] G. Karjoth, N. Asokan, C. Gülcü: Protecting the Computation Results of Free-Roaming Agents. Mobile Agents (MA'98), Second International Workshop, Proceedings, LNCS 1477, Springer Verlag, 1998

[KarKur04] Paul A. Karger, Helmuth Kurth: Increased information flow needs for high-assurance composite evaluations. 2nd IEEE International Information Assurance Workshop (IWIA'04), Proceedings, IEEE Computer Society Press, 2004

[KaSi99] P. Karn, W. Simpson. RFC2522: Photuris: Session Key Managment Protocol, URL:http://asg.web.cmu.edu/rfc/rfc2522.html, März, 1999

[KarOweO02] Karygiannis, T. Owens, L.: Wireless Network Security 802.11, Bluetooth and Handheld Devices. National Institute of Standards and Technology Special Publication 800-48, November 2002

[KohNeu93] J. Kohl, C. Neuman: The Kerberos Network Authentication Service (V5). IETF, RFC 1510, 1993

[KohHar2002] Ruyji Kohno and Shinichiro Haruyama: Software Radio in Japan, In "Software Defined Radio: Origins, Drivers and International Perspectives", Edited by Walter Tuttlebee, John Wiley & Sons, LTD, 2002.

[Konh00] Walter Konhaeuser, Innovating the Mobile World beyond the Third Generation, Proc. Visions of the Wireless World, Workshop of the Wireless Strategic Initiative, December 12, 2000, Brussels, Belgium

[KJJ99] Paul Kocher, Joshua Jaffe, Benjamin Jun: Differential power analysis: Leaking secrets. Crypto 99, Proceedings, LNCS 1666, Springer Verlag, 1999

[KnoRöh00] Konstantin Knorr, Susanne Röhrig: Security of Electronic Business Applications: Structure and Quantification. Electronic Commerce and Web Technologies (EC-Web 2000), LNCS 1875, Springer Verlag, 2000

[Kocher96] Paul C. Kocher: Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology – Crypto 1996, LNCS 1109, Springer Verlag, 1996

[Kranenburg03] Herma van Kranenburg: 4G+ - 4th Generation Platform Launching Ubiqitous Services. 4G+ Presentation, January 2003

[KSEE+03] Herma van Kranenburg, Alfons Salden, Henk Eertink, Ronald van Eijk, Johan de Heer: Ubiquitous Attentiveness — enabling context-aware mobile applications and services. 1st European Symposium on Ambient Intelligence (EUSAI 2003), Proceedings, LNCS 2875, Springer Verlag, 2003

[KSI04] Satoshi Kondo, Shinsuke Suzuki, Atsushi Inoue: Quarantine Model Overview for IPv6 Network Security. draft-kondo-quarantine-overview-01, July 2004

[LA03a] Liberty Alliance: Identity Systems and Liberty Specification Version1.1 Interoperability. Technical White Paper, February 2003

[LA03b] Liberty Alliance: Introduction to the Liberty Alliance Identity Architecture. White Paper, March 2003

[LA03c] Liberty Alliance: Business Benefits of Federated Identity. White Paper, February 2003

[Landau03] Susan Landau: Liberty ID-WSF Security and Privacy Overview. www.projectliberty.org, 2003

[Law01] J. Law, and K. Nomura, “NTT Communications’ Global IPv6 Strategy”, European Commission IPv6 Task Force 1st Phase Final Report

[LBFM+93] Bev Littlewood, Sarah Brocklehurst, Norman Fenton, Peter Mellor, Stella Page, David Wright, John Dobson, John McDermid, Dieter Gollmann: Towards operational measures of computer security. Journal of Computer Security, Vol. 2, No. 2-3, 1993

[LenVer01] Arjen K. Lenstra, Eric R. Verheul: Selecting cryptographic key sizes. Journal of Cryptology, Vol. 14, No. 4, 2001

[Lewis04] Norman Lewis: From customization to ubiquitous personalization: digital identity and ambient network intelligence. interactions, Vol. 11, No. 2, March / April, 2004

[LGGV00] C. de Laat, G. Gross, L. Gommans, J. Vollbrecht, D. Spence. Generic AAA Architecture. RFC 2903, August, 2000

[Linn03] John Linn: Liberty Trust Models Guidelines, Version 1.0. www.projectliberty.org, 2003

[LiuRic00] Chang Liu, Debra Richardson: Automated Security Checking and Patching Using TestTalk. Proceedings of the Fifteenth IEEE International Conference on Automated Software Engineering (ASE'00), September 2000

[LKH04] Shou-Chuan Lai, Wen-Chu Kuo, Mu-Cheng Hsieh: Defending against Internet Worm-like Infestations. 18th International Conference on Advanced Information Networking and Applications (AINA'04), Proceedings, 2004

[Loh2002] T. Lohmar, et al., "Interactive Broadcast Services within IMT-2000 and Beyond Systems", WWRF7, Dec. 2002

[LTSK+03] Arjen K. Lenstra, Erjan Trommer, Adi Shamir, Will Kortsmit, Bruce Dodson, James Hughes, Paul Leyland: Factoring estimates for a 1024-bit RSA modulus. ASIACRYPT'03, LNCS 2894, Springer Verlag, 2003

[Linn00] Trust Models and Management in Public-Key Infrastructures, RSA Laboratories, November 2000

[Linn03] John Linn: Liberty Trust Models Guidelines, Version 1.0. www.projectliberty.org, 2003

[M.1078] ITU-R Recommendation M.1078. Security Principles For IMT-2000

[M.1223] ITU-R Recommendation. Evaluation Of Security Mechanisms For IMT-2000, 1997

[MA02] Mishra, A., Arbaugh, W.A.: An Initial Security Analysis of the IEEE 802.1X Standard. http://www.drizzle.com/~aboba/IEEE/802-1x-d11.pdf February 2002

[MaHF03] A. Malpani, R. Housley, T. Freeman. Simple Certificate Validation Protocol (SCVP). Internet Draft, June, 2003

[Mar00] Martius, Kai: Sicherheitsmanagement in TCP/IP-Netzen, Vieweg Verlag, Januar 2000

[MalHouFre03] A. Malpani, R. Housley, T. Freeman: Simple Certificate Validation Protocol (SCVP). Internet Draft, 2003.

[MAMG99] M. Myers, R. Ankney, A. Malpani, S. Galperin, C. Adams. X.509 Internet Public Key Infrastructure Online Certificate Status Protocol — OCSP. RFC 2560, June, 1999

[MAM+99] M. Myers, R. Ankney, A. Malpani, S. Galperin, C. Adams: Internet X.509 Public Key Infrastructure: Online Certificate Status Protocol – OCSP. RFC 2560, June 1999

[Mc02] Microsoft Corporation: Windows XP Wireless Deployment Technology and Component Overview, November 2002

http://www.microsoft.com/technet/prodtechnol/winxppro/maintain/wificomp.mspx

[MD98] Madson C.; Doraswamy, N.:The ESP DES-CBC Cipher Algorithm with Explicit IV. Request for Comments 2405, November 1998

[MenOorMen96] Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone: Handbook of Applied Cryptography. CRC Press, 1996

[Mercuri03] Rebecca T. Mercuri: Analyzing Security Costs. Communications of the ACM, Vol. 46, No. 6, June 2003

[MeyRak03] Sven Meyer, Andry Rakotonirainy: A Survey of Research on Context-Aware Homes. Australian Workshop on Wearable, Invisible, Context-aware, Ambient, Pervasive, and Ubiquitous Computing, February 2003

[MG98a] Madson C.; Glenn, R.: The Use of HMAC-MD5-96 within ESP and AH. Request for Comments 2403, November 1998

[MG98b] Madson C.; Glenn, R.: The Use of HMAC-SHA-1-96 within ESP and AH. Request for Comments 2404, November 1998

[MGVT04] Bharat B. Madan, Katerina Goseva-Popstojanova, Kalyanaraman Vaidyanathan, Kishor S. Trivedi: A method for modeling and quantifying the security attributes of intrusion tolerant systems. Performance Evaluation, Vol. 56, Number 1-4, March 2004,

[MHF03] A. Malpani, R. Housley, T. Freeman: Simple Certificate Validation Protocol (SCVP). Internet Draft, October 2003

[MilHol03] Lynett I. Millet, Stephen H. Holden: Authentication and its Privacy Effects. IEEE Internet Computing, Vol. 7, No. 6, 2003

[Mih2002] A. Mihailovic, et al., “Aspects of Multi-Homing in IP Access Networks”, IST Mobile Summit 2002

[Min00] S. Mink, F. Pählke, G. Schäfer, and J. Schiller, “Towards Secure Mobility Support for IP Networks”, International Conference on Communication Technologies (ICCT 2000), August 2000, Beijing, China

[Mink] S.Mink et.al., “Security aspects of micro-mobility supporting architectures for IP networks,” presentation in IPCN 2000, May 2000.

[Mink2000] Stefan Mink, Frank Pählke, Günter Schäfer, Jochen Schiller, “Towards Secure Mobility Support for IP Networks”, 2000

[MilHol03] Lynett I. Millet, Stephen H. Holden: Authentication and its Privacy Effects. IEEE Internet Computing, Vol. 7, No. 6, 2003

[MIP NAT] Mobile IP NAT/NAPT Traversal using UDP Tunnelling <draft-ietf-mobileip-nat-traversal-07.txt>

[MIP-Threats] Mobile IP: Introduction, Security issues http://staff.cs.utu.fi/kurssit/computer_and_network_security/spring_2000/MIP.pdf

[MIPv4 Trav] Mobile IPv4 Traversal Akreuz IPsec-based VPN Gateways <draft-ietf-mobileip-vpn-problem-solution-00>

[MiShSt98] John C. Mitchell, Vitaly Shmatikow, Ulrich Stern: Finite State Analysis of SSL 3.0, University of Stanford, 1998

[MMM00] Akio Moridera, Kazuo Murano, Yukou Mochida: The Network Paradigm of the 21st Century and Its Key Technologies. IEEE Communications Magazine, Vol. 38, No. 11, 2000

[MoiKon00] Soumyo D. Moitra, Suresh L. Konda: The survivability of network systems: an empirical analysis. Carnegie Mellon University, Software Engineering Institute, Technical Report CMU/SEI-2000TR-021, ESC-TR-2000-021, December 2000

[moac] mobileaccess.de – public spots in deiner nähe: mobileaccess.de/wlan

[Mohr] W. Mohr and W. Konhauser, "Access network evolution beyond third generation mobile communications," IEEE Comm. Mag., December 2000, pp. 120-132.

[Moi00] Moioli, F. Msc.: Security in Public Access Wireless Lan Networks, Thesis

[Montavont] N. Montavont and T. Noel, "Handover management for mobile nodes in IPv6 networks," IEEE Comm. Mag., Aug. 2002, pp. 38-43.

[Moore65] Gordon E. Moore: Cramming more components onto integrated circuits. Electronics, Volume 38, Number 8, April 1965

[MorMurMoc00] Akio Moridera, Kazuo Murano, Yukou Mochida: The Network Paradigm of the 21st Century and Its Key Technologies. IEEE Communications Magazine, Vol. 38, No. 11, 2000.

[MSServ03] Microsoft Corporation: Microsoft Windows Server 2003 Network Access Quarantine Control. October 2003, http://www.microsoft.com/

[MSServ04] Microsoft Corporation: Introduction to Network Access Protection for Microsoft Windows Server 2003. June 2004, http://www.microsoft.com/windowsserver2003/technologies/networking/nap/default.mspx

[MSVS03] David Moore, Colleen Shannon, Geoffrey Voelker, Stefan Savage: Internet Quarantine: Requirements for Containing Self-Propagating Code. INFOCOM 2003, Proceedings

[MSST98] D. Maughan, M. Schertler, M. Schneider, J. Turner: Internet Security Association and Key Management Protocol. RFC 2408, November 1998

[MSS+98] Maughan, D.; Schertler, M.; Schneider, M.; Turner, J.: Internet Security Association and Key Management Protocol (ISAKMP). Request for Comments 2408, November 1998

[MTT] C. Prehofer: "Mobile Terminal Technology: Trends and Future Development", DoCoMo Eurolabs, Dagstuhl 2002

[MyeAnkMalGalAda99] M. Myers, R. Ankney, A. Malpani, S. Galperin, C. Adams: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol — OCSP. RFC 2560, 1999.

[NAT Comp1] Protocol Complications with IP Network Address Translation, RFC 3027.

[NAT Comp2] IPSec NAT Compatibility Requirements, draft-ietf-ipsec-nat-reqts-03.txt

[NAT Trav] Negotiation of NAT Traversal in the IKE, draft-ietf-ipsec-nat-t-ike-05.txt.

[NAT trouble] Trouble with NAT, Cisco Internet Protocol Journal, December 2000.

[Noll at. al. 2001] Josef Noll, Richard Dennis, Jaime Ferreira, Michael Barry, and Stein Svaet: Concepts for the Roadmap beyond 3G; In Conference Proceeding of EURESCOM Summit 2001 pp. 241 – 251

[Nück06] Nückel, Armin: Mobliity requires Security, 2006

[NW03] Network World: Windows patch management tools, 2003, http://www.nwfusion.com/reviews/2003/0303patchrev.html

[ODK99] Rodolphe Ortalo, Yves Deswarte, Mohamed Kaâniche: Experimenting with Quantitative Evaluation Tools for Monitoring Operational Security. IEEE Transactions on Software Engineering, Vol. 25, No. 5, September/October, 1999

[Ojanpera] T. Ojanpera and R. Prasad, WCDMA: Towards IP Mobility and Mobile Internet, Artech House, Norwood, USA, April 2001.

[OSGi] OSGi. http://www.osgi.org

[Pahalavan] K. Pahalavan et.al., "Handoff in hybrid mobile data networks," IEEE Pers. Comm., Apr. 2000, pp. 34-47.

[PalmOSAR] http://www.palmsource.com/fr/enterprise/Palm_OS5_Overview.pdf

[PAK+01] Patel, Baiju; Aboba, Bernard; Kelly, Scott; Gupta, Vipul: DHCPv4 Configuration of IPsec Tunnel Mode. Internet Draft <draft-ietf-ipsec-dhcp-13.txt>, 2001

[PAMPAS] European Project, Title: Pioneering Advanced Mobile Privacy and Security. http://www.pampas.eu.org/

[PANA] PANA-WG homepage, http://www.ietf.org/html.charters/pana-charter.html

[Paulson97] Lawrence C. Paulson: Inductive Analysis of the Internet Protocol TLS, University of Cambridge, 1997

[PasMit03] Andreas Pashalidis, Chris J. Mitchell: A Taxonomy of Single Sign-On Systems. Information Security and Privacy, 8th Australasian Conference (ACISP 2003), LNCS 2727, Springer Verlag, July, 2003.

[PasMit03] Andreas Pashalidis, Chris J. Mitchell: A taxonomy of SSO systems. ACISP 2003, LNCS 2727, Springer Verlag, 2003

[passOne] pass-One Homepage: www.pass-one.com, March 2003

[PDC] S. Hirata et.al. "PDC mobile packet data communications network", IEEE, 1995, pp. 644-648.

[PDC_PDC-P] http://www.mobilecomms-technology.com/projects/pdc/

[Per01] J. Pereira, "Increasing Spectrum Efficiency: Leitmotiv for FP6 MMC'01", Nov. 2001, Berlin, Germany, http://www.ist-drive.org/MMC2001/3.1_Spectrum.pdf

[Per99] Perkins, C.: Mobile IP and security issue: an overview. Proceedings of 1st IEEE Workshop on Internet Technologies and Services, 1999.

[PerMan03] Karl E Persson, D. Manivannan: Secure Connections in Bluetooth Scatternets. Proceeding of the 36th Hawaii International Conference on System Sciences , HICSS'03, 2003

[PET00] Petander, H.: Mobile IP Route Optimization. http://www.hut.fi/~lpetande/internetworking/R_Opt.html, 2000

[PGP] http://www.pgpi.org/

[PHSa] www.phsmou.org/resources/AdvancedPHS.pdf

[PHSb] PHS MoU Document A-GN0.00-01-TS, www.phsmou.org/resources/pdf/A-GN0.00-01-TS.pdf

[PHS97] Public Personal Handy-Phone System: Network and System Configurations, PHS MoU Document B-NW 1.00-04-TS, Dezember 1997, www.phsmou.org/resources/pdf/B-NW1.00-04-TS.pdf

[PHS_HO99] http://www.phsmou.or.jp/newsletter/issue21/handover.aspx

[PinHou02] D. Pinkas, R. Housley: Delegated Path Validation and Delegated Path Discovery Protocol Requirements. RFC 3379, 2002.

[Pip98] Piper, D.: The Internet IP Security Domain of Interpretation for ISAKMP. Request for Comments 2407, November 1998

[PJ00] Perkins,C. and Johnson,D.: Route Optimization in Mobile IP. Internet Draft, Februar 2000, http://www.monarch.cs.rice.edu/internet-drafts/draft-ietf-mobileip-optim-09.txt

[Plam87] Plamondon, R., G. Lorette : Automatic Signature Verification and Writer Identifikation – State of the Art in Pattern Recognition, Vol. 22,2: (1987)

[Pollini] G.P. Pollini, “Trends in handover design,” IEEE Comm. Mag., March 1996, pp. 82-90.

[Prasad] N.R. Prasad and A.R. Prasad, WLAN Systems and Wireless IP for Next Generation, Jan. 2002, Artech House, Norwood, USA.

[Prob MIPv4 Trav] Problem Statement: Mobile IPv4 Traversal of VPN Gateways <draft-ietf-mobileip-vpn-problem-statement-req-01>

[Q.1701] ITU-T Recommendation Q.1701. Framework for IMT-2000 networks

[RB98] Redi J., Bahl, P., Mobile IP: A Solution for Transparent Seamless Mobile Computer Communications. Fuji-Keizai's Report on Upcoming Trends in Mobile Computing and Communications, July 1998

[RBAC] http://csrc.nist.gov/rbac/

[RBMR+03] Natalia Romero, Joy van Baren, Panos Markopoulos, Boris de Ruyter, Wijnand Ijsselsteijn: Addressing Interpersonal Communication Needs through Ubiquitous Connectivity: Home and Away. 1st European Symposium on Ambient Intelligence (EUSAI 2003), Proceedings", LNCS 2875, Springer Verlag, 2003

[RFC-791] J. Postel (editor), “Internet Protocol”, RFC 791, Sept. 1981

[RFC-792] J. Postel, “Internet Control Message Protocol”, STD 5, RFC 792, Sept. 1981

[RFC-1305] D. Mills, “Network Time Protocol (Version 3) Specification, Implementation”, RFC 1305, March 1992

[RFC-1981] J. McCann, S. Deering, and J. Mogul, “Path MTU Discovery for IP version 6”, RFC 1981, Aug. 1996

[RFC-2002] C. Perkins: IP Mobility Support, RFC 2002, Oktober 1996

[RFC-2003] C. Perkins: IP Encapsulation within IP, RFC-2003 Oktober 1996

[RFC-2373] R. Hinden, and S. Deering, “IP Version 6 Addressing Architecture”, RFC 2373, Jul. 1998

[RFC-2401] Kent, S. and R. Atkinson, “Security Architecture for the Internet Protocol”, RFC 2401, Nov. 1998

[RFC-2402] Kent, S. and R. Atkinson, “IP Authentication Header”, RFC 2402, Nov. 1998

[RFC-2406] Kent, S. and R. Atkinson, “IP Encapsulating Security Protocol (ESP)”, RFC 2406, Nov. 1998

[RFC-2460] S. Deering, and R. Hinden, “Internet Protocol, Version 6 (IPv6) Specification”, RFC 2460, Dec. 1998

[RFC-2461] T. Narten, E. Nordmark, and W. Simpson, “Neighbor Discovery for IP Version 6 (IPv6)”, RFC 2461, Dec. 1998

[RFC-2463] A. Conta, and S. Deering, “ICMP for the Internet Protocol Version 6 (IPv6)”, RFC 2463, Dec. 1998

[RFC-2784] D. Farinacci; T. Li; S. Hanks; D. Meyer; P. Traina: Generic Routing Encapsulation (GRE), RFC 2784, März 2000.

[RFC-2977] S. Glass, et al, “Mobile IP Authentication, Authorization, and Accounting Requirements”, RFC2977, Oct. 2000

[RFC-3344] C. Perkins (editor), IP Mobility Support for IPv4, RFC 3344, Aug. 2002

[Ricardo et al. 2002] M. Ricardo, J. Dias, G. Carneiro, J. Ruela UMTS Terminal Equipment For All-IP Based Communications http://citeseer.ist.psu.edu/558710.html

[Richardson03] Robert Richardson: 2003 CSI / FBI Computer Crime and Security Survey. Computer Security Institute, www.gocsi.com, 2003

[RKVP+03] Katja Rentto, Ilkka Korhonen, Antti Väätänen, Lasse Pekkarinen, Timo Tuomisto, Luc Cluitmans, Raimo Lappalainen: Users' Preferences for Ubiquitous Computing Applications at Home. 1st European Symposium on Ambient Intelligence (EUSAI 2003), Proceedings, LNCS 2875, Springer Verlag, 2003

[Roth01] Volker Roth: On the robustness of some cryptographic protocols for mobile agent protection. 5th International Conference on Mobile Agents (MA 2001), Proceedings, LNCS 2240, Springer Verlag, 2001

[Roth02] Jörg Roth: Mobile Computing: Grundlagen, Technik, Konzepte; dpunkt Verlag Heidelberg, 2002

[RSA78] Ron L. Rivest, Adi Shamir, Leonard M. Adleman: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM, Vol. 21, No. 2, February 1978

[RSA02] RSA Security: RSA SecureID Authentication – A better value for a better ROI. White Paper, 2002

[RSA02] RSA Security: RSA SecureID. Product Information, 2004

[RSCJ+02] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler: SIP: Session Initiation Protocol. RFC 3261, June 2002

[RSIP1] Realm Specific IP: A Framework, RFC 3102.

[RSIP2] Realm Specific IP: Protocol Specification, RFC 3103.

[RSIP3] RSIP Support for End-to-End IPsec, RFC 3104.

[Rubin95] Aviel D. Rubin: Independent One-Time Passwords. Proceedings of the 5th USENIX Security Symposium, June 1995

[SAGE] ETSI Security Algorithms Group of Experts (SAGE): Report on the specification, evaluation and usage of the GSM GPRS Encryption Algorithm (GEA). ETSI TR 101 375 V1.1.1, 1998

[Sam99] Sami, P., Mobile IP. http://www.tml.hut.fi/Studies/Tik-110.300/1999/Essays/MIP.html#luku5, 1999

[SAML02a] OASIS: Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML). OASIS Standard, oasis-sstc-saml-core-1.0, (www.oasis-open.org), November, 2002.

[SAML02b] OASIS: Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML). OASIS Standard, oasis-sstc-saml-sec-consider-1.0, (www.oasis-open.org), November, 2002.

[SAML02c] OASIS: Conformance Program Specification for the OASIS Security Assertion Markup Language (SAML). OASIS Standard, oasis-sstc-saml-bindings-1.0, (www.oasis-open.org), November, 2002.

[SAML02d] OASIS: Bindings and Profiles for the OASIS Security Assertion Markup Language (SAML). OASIS Standard, oasis-sstc-saml-bindings-1.0, (www.oasis-open.org), November, 2002.

[SBSH+03] Marianne Swanson, Nadya Bartol, John Sabato, Joan Hash, Laurie Graffo: Security Metrics Guide for Information Technology Systems. NIST — National Institute of Standards and Technology, NIST Special Publication 800-55", July 2003

[Schiller03] Jochen Schiller: Mobilkommunikation, Pearson Education, München, 2003

[SchWed00] Henning Schulzrinne, Elin Wedlund: Application-layer mobility using SIP. ACM SIGMOBILE Mobile Computing and Communications Review, Vol. 4, No. 3, July 2000

[Schneier96] Bruce Schneier: Angewandte Kryptographie, Addison Wesley, 1996

[Schneier99] Bruce Schneier: Attack Trees – Modelling Security Threats. Dr. Dobb's Journal, December 1999

[SDRF_Sec2002] Report on Issues and Activity in the Area of Security for Software Defined Radio 1 September 2002 !!!!

[SDRF_SecArch] SDR System Security S&A-SEC Document No. SDRF-02-A-0006-V0.00 !!!!

[Ses1997] Srinivasan Seshan, et al., "Handoffs in Cellular Wireless Networks: The Daedalus Implementation and Experience", Kluwer International Journal on Wireless Personal Communications, Jan. 1997

[SGF02] Gary Stonebumer, Alice Goguen, Alexis Feringa: Risk Management Guide for Information Technology Systems. NIST — National Institute

of Standards and Technology, NIST Special Publication 800-30", July 2002

[Shannon49] Claude E. Shannon: Communication theory of secrecy systems. Bell System Technical Journal, Vol. 28, No. 4, October 1949

[ShaTro03] Adi Shamir, Eran Tromer: Factoring large numbers with the TWIRL device. Advances in Cryptology — CRYPTO'03, LNCS 2729, Springer Verlag, 2003

[SHAMAN01] IST-2000-25350 – SHAMAN, D02.2, "Intermediate Report: Results of Review,Requirements and Reference Architecture," 29-June-01.

[SHAMAN00] European Project, Title: Security for Heterogeneus Access in Mobile Applications and Networks. http://www.ist-shaman.org/

[SHAMAN02] R. Schmitz (editor), "Intermediate Report: Results of Review, Requirements and Reference Architecture", IST project SHAMAN Deliverable D02, Jun. 2002

[ShaTro03] Adi Shamir, Eran Tromer: Factoring large numbers with the TWIRL device. Advances in Cryptology — CRYPTO'03, LNCS 2729, Springer Verlag, 2003

[Silverman00] Robert D. Silverman: A cost-based security analysis of symmetric and asymmetric key lengths. RSA Bulletin #13, April 2000 (revised November 2001)

[SLI00] Evdoxia Spyropoulou, Timothy Levin, Cynthia Irvine: Calculating costs for quality of security service. 16th Annual Computer Security Applications Conference (ACSAC 2000), Proceedings, IEEE Computer Society Press, 2000

[SOAP1.1] Don Box, David Ehnebuske, Gopal Kakivaya, Andrew Layman, Noah Mendelsohn, Henrik Frystyk Nielsen, Satish Thatte, Dave Winer: Simple Object Access Protocol (SOAP) 1.1. W3C Note, May, 2000.

[Someren02] Nicko van Someren: The risks of short RSA keys for secure communication using SSL. White Paper, nCipher, April 2002

[Spurling01] Kevin W. Spurling: Home Networking -- a comparison of modern technologies. University of Maryland, INSS 690, September 2001

[Spu] Sputnik Project: www.sputnik.com

[Sri99] Srisuresh, P.: Security Model with Tunnel-mode IPsec for NAT Domains. Request for Comments 2709, October 1999

[SSL] http://wp.netscape.com/eng/ssl3/

[SSL3] Alan O. Freier, Philip Karlton, Paul C. Kocher: The SSL Protocol Version 3.0, draft-freier-ssl-version3-02.txt, 18. November 1996

[SSE04] H. Sarbinowski, T. Shafi, C. Eckert: ESI – der elektronische Sicherheitsinspektor. IT-Sicherheit, Februar 2004

[Steinm99] Steinmetz, R.: Multimedia Technologie Grundlagen Komponenten und Systeme, Springer Verlag, Berlin, Heidelberg, 1999

[Str] Michael Strauss, PASS-Consulting Group:Historie von UMTS,http://www.pass-consulting.com/mobile/html_d/technologies/pdf/umts.pdf

[SUN] http://java.sun.com/products/midp/

[Symbian] http://www.symbian.com

[symbianOSAR] http://www.symbian.com/technology/symbos-v7x-det.html

[Szacik01] Robert S. Szacik: HomeRF: Wireless with Security, for the Rest of Us? , 2001, http://www.giac.org/practical/gsec/Bob_Szacik_GSEC.pdf

[Tanen98] Tanenbaum, A.: Computernetzwerke, Prentice Hall, 1998

[Taylor] L. taylor, “The challenges of seamless handover in future mobile multimedia networks,” IEEE Pers. Comm., April 1999, pp. 32-37.

[Tele01] TeleTrust Deutschland e.V., Arbeitsgruppe 6: Biometrische Identifikationsverfahren. Bewertungskriterien zur Vergleichbarkeit biometrischer Verfahren. Kriterienkatalog, 2001-08-25

[ThEM03] Mary R. Thompson, Abdelilah Essiari, Srilekha Mudumbai. Certificate-based Authorization Policy in a PKI Environment. ACM Transactions on Information and System Security, Vol. 6, No. 3, August, 2003

[ThoEssMud03] Mary R. Thompson, Abdelilah Essiari, Srilekha Mudumbai: Certificate-based Authorization Policy in a PKI Environment. ACM Transactions on Information and System Security, Vol. 6, No. 3, August, 2003.

[TJMH99] Mary Thompson, William Johnston, Srilekha Mudumbai, Gary Hoo, Keith Jackson, Abdelilah Essiari. Certificate-Based Access Control for Widely Distributed Ressources. 8th USENIX Security Symposium, Proceedings, August, 1999

[TIKM+02] Eiji Tokunaga, Hiro Ishikawa, Makoto Kurahashi, Yasunobu Morimoto, Tatsuo Nakajima: A Framework for Connecting Home Computing Middleware. 22nd International Conference on Distributed Computing Systems Workshops (ICDCSW '02), Proceedings, IEEE Computer Society Press, July 2002

[Timm99] Paul Timmers: Electronic Commerce, Wiley, 1999

[TMEC02] Mary R. Thompson, Srilekha Mudumbai, Abdelilah Essiari, Willie Chin. Authorization Policy in a PKI Environment. Proceedings of the 1st Annual NIST Workshop on PKI, April, 2002

[T-Mobile] T-Mobile Hotspots Homepage: http://www.t-mobile.com/hotspot

[TR45.2] "T1-TIA Coordination Group: IMT-2000 Security Requirements: Capability Set 1"",Subcommittee TR-45.2, Dec. 98"

[Tripathi] N.D. Tripathi et.a., "Handoff in cellular systems, " IEEE Pers. Comm., Dec. 1998, pp. 26-37.

[TSS99] Tuquerres, G., Salvador, M. R., Sprenkels, R., Mobile IP: Security & Application. http://ganges.cs.tcd.ie/htewari/papers/, 1999

[Tuesday03] Vince Tuesday: Bad policy makes for weak passwords. Computerworld, December 2003

[TsuSun03] Tak-Goa Tsuan, Chih-Yang Sung: Ubiquitous information services with JAIN platform. Mobile Networks and Applications, Vol. 8, No. 6, Kluwer Academic Publishers, December 2003

[Tuttlebee02 1] Software Defined Radio: Origins, Drivers and International Perspectives; Edited by Walter Tuttlebee; John Wiley & Sons Ltd 2002

[umtsov] http://www.umtsworld.com/technology/overview.htm

[UMTSlink] http://umtslink.at

[umtsse] http://www.umtsworld.com/technology/security.htm

[UMTSworld] http://www.umtsworld.com/technology/handover.htm

[Usi02] 3GPP TS 21.111: USIM and IC card requirements. V5.1.0, 2002

[Van96] Geneviève Vanneste, et al., Initial report on security requirements, ACTS project ASPeCT Deliverable AC095/ATEA/W21/DS/P/02/B.2, Feb.1996

[Varney03] Christine Varney: Privacy and Security Best Practices, Version 2.0. www.projectliberty.org, 2003

[Varian00] Hal R. Varian: Managing Online Security Risks. http: // www.nytimes.com, 2000

[VauHen02] Ray Vaughn, Ronda Henning: A Report on the Information System Security Rating and Ranking Workshop. 14th Annual Canadian Information Technology Security Symposium, May 2002

[VCFG00] J. Vollbrecht, P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, D. Spence. AAA Authorization Application Examples. RFC 2905, August, 2000

[VCFG99] Vollbrecht, J.; Calhoun P.; Farrell, S.; Gommans, L.; Gross, G.; de Bruijn, B.; Holdrege, M.; Spence, D.: AAA Authorisation Architecture and Requirements. Internet-Draft, 1999

[ViMeCh02] John Viega, Matt Messier, Pravir Chandra: Network Security with OpenSSL, O'Reilly, Juni 2002

[Vision2000] WWRF, “The Book of Visions 2000, Visions of the Wireless World”, Nov. 2000

[Vin98] Bart Vinck, et al., Overview of UMTS architecture, ACTS project USECA Deliverable D12, June 1998; http://www.eurescom.de/public/projects/p1000-series/P1046

[VMKA03] Elena Vildjiounaite, Esko-Juhani Malm, Jouni Kaartinen, Petteri Alahuhta: Context Awareness of Everyday Objects in a Household. 1st European Symposium on Ambient Intelligence (EUSAI 2003), Proceedings, LNCS 2875, Springer Verlag, 2003

[Wagner96] David Wagner, Bruce Schneier: Analysis of the SSL 3.0 Protocol, University of Berkley, 1996

[Wagner97] David Wagner, Bruce Schneier: Analysis of the SSL 3.0 Protocol, Revised, University of Berkley, April 1997

[Walke01a] Walke, Bernhard: Mobilefunknetze und ihre Protokolle Band 1. Teubner-Verlag, 3.Auflage, 2001

[Walke01b] Walke, Bernhard: Mobilefunknetze und ihre Protokolle Band 2. Teubner-Verlag, 3.Auflage, 2001

[Wang] H. Wang, “Summary of security mechanisms in nowadays mobile communication technologies/systems,” SBSS meeting, 21 Nov. 2002, Munich, Germany.

[Wason03] Thomas Wason: Liberty ID-FF Architecture Overview, Version 1.2. www.projectliberty.org, 2003

[WATSON] http://www.physics.udel.edu/wwwusers/watson/scen103/intel.html adapted from the Microprocessor Report 9(6), Mai 1995

[WanWul97] Chenxi Wang, William A. Wulf: Towards a framework for security measurement. 20th National Information Systems Security Conference (NISSC'97), October 1997

[Wea00] Weatherspoon, S., Overview of IEEE 802.11b Security, 2000 http://www.intel.com/technology/itj/archive/2000.htm

[WedSch99] Elin Wedlund, Henning Schulzrinne: Mobility Support using SIP. 2nd ACM/IEEE International Workshop on Wireless and Mobile Multimedia (WoWMoM'99), 1999

[Weiser93] Mark Weiser: Some Computer Science Issues in Ubiquitous Computing. Communications of the ACM, Vol. 36, No. 7, July 1993

[Wessung01] Henrik Wessung: Home Access Technology. Uema University Sweden, October 2001

[WFCR01] Huaqiang Wei, Deb Frinke, Olivia Carter, Chris Ritter: Cost-Benefit Analysis for Network Intrusion Detection Systems. CSI 28th Annual Computer Security Conference, Proceedings, October 2001

[Wi-Fi] Wi-Fi Homepage http://www.wirelessfidelity.org/

[Wis02] Wiese, H.: Das neue Internetprotokoll IPv6. Carl Hanser Verlag, 2002

[Wiener93] Michael Wiener: Efficient DES key search. Presented at the rump session of Crypto ’93, 1993

[WordNet04] WordNet Dictionary, http://wordnet.princeton.edu, September 2004

[WUSB05a] Agere Systems, Hewlett Packard Company, Intel Corporation, Microsoft Corporation, NEC Corporation, Koninlijke Philips Electronics N.V., Samsung Electronics: Wireless Universal Serial Bus Specification, Revision 1.0, Mai 2005

[WUSB05b] Agere Systems, Hewlett Packard Company, Intel Corporation, Microsoft Corporation, NEC Corporation, Koninlijke Philips Electronics N.V., Samsung Electronics: Wireless Universal Serial Bus Specification, Errata on Revision 1.0, Juli 2005

[WUSB06] Agere Systems, Hewlett Packard Company, Intel Corporation, Microsoft Corporation, NEC Corporation, Koninlijke Philips Electronics N.V., Samsung Electronics: Association Models Supplement to the Certified Wireless Universal Serial Bus Specification, Revision 1.0, März 2006

[WWRF7] A.R. Prasad and P. Schoo, “IP Security for beyond 3G towards 4G,” WWRF#7, December 3-4, 2002, Eindhoven, The Netherlands.

[WWSB+04] Cynthia Wong, Chenxi Wang, Dawn Song, Stan Bielski, Gregory Ganger: Dynamic Quarantine of Internet Worms. International Conference on Dependable Systems and Networks (DSN'04), Proceedings

[X.509] ITU-T Recommendation X.509 (1997 E): Information Technology − Open Systems Interconnection − The Directory: Authentication Framework. June 1997.

[XuHis02] B. Xu, and S. Hischke, “An IP-based Broadband Wireless Access Network Architecture supporting Ad hoc Networking Integration”, Proceedings of EURESCOM Summit 2002, Oct. 2002, Heidelberg, Germany

[Yeg02] A. E. Yegin, “Secure Network Access”, Mar. 2002, http://www.connectathon.org/talks02/alper.pdf

[Yee01] Bennet S. Yee: Security Metrology and the Monty Hall Problem. Workshop on Information Security System Rating and Ranking, May 2001

[Yu00] Yu, S.: Security Mechanisms and Countermeasures in the mobile IP Environment, 2000 http://fiddle.visc.vt.edu/courses/ecpe6504-wireless/projects_spring2000/pres_yu.pdf

[ZC97] Zao, John K., Condell, M., Internet Draft - Use of IPsec in Mobile IP, http://www3.ietf.org/proceedings/98aug/I-D/draft-ietf-mobileip-ipsec-use-00.txt , November 1997

[ZGT+99] Zao, J., Gahm, J., Troxel, G., Condell, M., Helinek, P., Yuan, N., Castineyra, I., Kent, S.: A public-key based secure Mobile IP. Wireless Networks, October 1999

[ZGT03] Changchun Zou, Weibo Gong, Don Towsley: Worm Propagation Modeling and Analysis under Quarantine Defense. WORM´03, October 2003

[ZSP01] S. Zhou, A. Seneviratne, T. Percival: A Location Management Scheme for Mobility Support in Wireless IP Networks Using Session Initiation Protocol (SIP). 9th IEEE International Conference on Networks (ICON'01), Proceedings, IEEE Computer Society Press, 2001

14 Abkürzungen

3GPP	3rd Generation Partnership Project
4GPlus	4th Generation Platform Launching Ubiquitous Services
AAA-Server	Authentifizierungs-, Autorisierungs- und Abrechnungs-Server
AC	Authentifikations-Zentrum
ACL	Access Control List
AES	Advanced Encryption Standard
AH	Authentication Header
AP	Access point
ARM	Advanced Risc Machine
ARP	Address Resolution Protocol
ARPA	Advanced Research Project Agency
AV	Authentication Vector
BS	Base Station
BSI	Bundesamt für die Sicherheit in der Informationstechnik
BU	Binding Update
BWA	Broadband Wireless Access
CA	Certification Authority - Zertifizierungsstelle
CAVE	Cellular Authentication and Voice Encryption
CBC Mode	Cipher Block Chaining-Modus
CCMP	Counter Mode with Cipher Block Chaining Message Authentication Code Protocol
CDMA	Code Division Multiple Access
CEPT	Conférence Européenne des Administrations des Postes et des Télecommunications
CERT	Computer Emergency Response Team
CFB	Cipher Feedback Modus
CGI	Common Gateway Interface
CHAP	Challenge Handshake Athentication Protocol
CN	Correspondent Node
CoA	Care-of-Address
CRL	Certificate Revocation List, Zertifikate Sperrliste
DARPA	U.S. Defense Advanced Research Project Agency
DCSSI	Direction Centrale de la Sécurité des Systèmes d'Information
DDoS	Distributed Denial of Service
DECT	Digital Enhanced/European Cordless Telecommunications
DES	Digital Data Encryption Standard
DH	Diffie-Hellmann
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name Service
DNSSec	Domain Name Service Security
DoS	Denial of Service

DFS	Dynamic Frequency Selection
DSA	Digital Signature Algorithm
DSC	DECT Standard Cipher
DSL	Digital Subscriber Line
DSS	Digital Signature Standard
EAP	Extensible Authentication Protocol
ECB	Electronic Code Block
ECC	Elliptic Curve Cryptography
EDGE	Enhanced Data Rates for GSM Evolution
EEPROM	Electronically Erasable Programmable Read Only Memory
ESP	Encapsulation Security Payload
ETSI	European Telecommunication Standards Institute
FA	Foreign Agent - Fremdagent
FDD	Frequency Division Duplex
FDMA	Frequency Division Multiplex Access
FN	Foreign Network - Fremdnetz
GEA	GPRS Encryption Algorithm
GERAN	GSM EDGE Radio Access Network
GGSN	Gateway GPRS Support Node
GPRS	General Racket Radio Service
GRE	Generic Routing Encapsulation
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
HA	Home Agent - Heimatagent
HE	Home Environment
HLR	Home Location Register
HMAC	Hashed Message Authentication Code
HN	Home Network - Heimnetz
HSCSD	High Speed Circuit Switched Data
HTML	Hypertext Markup Language
HTTP	Hypertext Transport Protocol
IAPP	Inter Access Point Protocol
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System - Einbruchserkennungs-System
IEEE	The Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IIS	Internet Information Service
IKE	Internet Key Exchange
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
IMT-2000	International Mobile Telecommunications-2000
IP	Internet Protocol
IPRA	Internet Policy Registration Authority
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6

ITU	International Telecommunication Union
IPSec	IP Security
IrDA	Infrared Data Association
ISAKMP	Internet SA and Key Management Protocol
ISDN	Integrated Services Digital Network
ISM	Industrial, Scientific, Medical Frequenzband
ISP	Internet Service Provider Internet Zugangsanbieter
IS-95	amerikan. Analogon zu GSM
ITSEC	Information Technology Security Evaluation Criteria
IT-Systeme	Informationstechnische Systeme
IuK	Informations- und Kommunikationstechnik
IuKDG	Informations- und Kommunikationsdienstegesetz
LA	Location Area
LCP	Link Control Protocol
LDAP	Lightweight Directory Access Protocol
MAC	Message Authentication Code
MAC-Adresse	Medium Access Control
MAN	Metropolitan Area Network
MAXXtelecom	WiMAX von DBD in Heidelberg
MBWA	Mobile Broadband Wireless Access
MD5	Message Digest 5
MH	Mobile Host
MIC	Message Integrity Check
MIME	Multipurpose Internet Mail Extension Protocol
MIP	Mobile Internet Provider
MN	Mobile Node – Mobiles Endgerät
MSC	Mobile Services Switching Center
MSISDN	Mobile Station International ISDN Number
NAT	Network Address Translation Protocol
NIS	Network Information Service
NSA	National Security Agency
Oakley	Oakley Key Determination Protocol
OCSP	Online Certificate Status Protocol
OFB	Output Feedback Modus
OFDM	Orthogonal Frequency Division Multiplex
OFDMA	OFDM Access
OTP	One-Time Password
PAP	Password Authentication Protocol
PC	Personal Computer
PCA	Policy Certification Authority
PDA	Personal Digital Assistant
PEM	Privacy Enhanced Mail
PGP	Pretty Good Privacy
PHS	Personal Handyphone System
PIN	Personal Identification Number

PKCS	Public Key Cryptography Standards
PKI	Public Key Infrastruktur
PKIS	PKI Server
PmP-RiFu	Punkt-zu-Punkt-Richtfunk
POP	Post Office Protocol
PP	Protection Profile
PPP	Point-to-Point-Protocol
PS	Personal Station
PSK	Pre-Shared Key
PUK	PIN Unblocking Key
QN	Quarantine Network
QOS	Quality of Service
RA	Registration Authority
RAM	Random Access Memory
RBAC	Role Based Access Control
RC4	Ron's Code 4
RegTP	Regulierungsbehörde für Telekommunikation und Post
RNC	Radio Network Controller
ROM	Read only Memory
RPC	Remote Procedure Call
RSA	Rivest Shamir Aldeman
RSP	Roaming Service Provider
SA	Security Association
SAG	SIM Access Gateway
SAM	Security Accounts Manager
SAML	Security Assertions Markup Language
SCVP	Simple Certificate Validation Protocol
SGSN	Serving GPRS Support Node
SHA	Secure Hash Algorithm
SIG	Bluetooth Special Interest Group
SigG	Signaturgesetz
SIM	Subscriber Identity Module
SMTP	Simple Mail Transfer Protocol
SOAP	Simple Object Access Protocol
SOFDMA	Scalable OFDM Access
SPI	Security Parameter Index
SS	Subscriber Station
SSL	Secure Socket Layer
SSO	Single Signe On
TA	Trust Anchor
TC	Trust Center - Zertifizierungsstelle
TCP	Transmission/Transport Control Protocol
TCSEC	Trusted Computer System Evaluation Criteria
TDD	Time Division Duplex
TDMA	Time Division Multiple Access

T-DSL	Breitband DSL der T-Com
TFN	Tribe Flood Network
TKIP	Temporal Key Integrity Protocol
TLS	Transport Layer Security
TMSI	Temporary Mobile Subscriber Identity
TPC	Transmit Power Control
TPM	Trusted Platform Module
UBW	Ultra-Wideband
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
URI	Uniform Resource Identification
URL	Uniform Resource Locator
USB	Universal Serial Bus
VLR	Visitor Location Register
VOIP VoIP	Voice over IP
VPN	Virtual Private Network
WAN	Wide Area Network
WCDMA	Wideband Code Division Multiple Access
W-DSL	Wireless DSL
WEP	Wired Equivalent Privacy
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WLL	Wireless Local Loop
WMAN	Wireless Metropolitan Area Network
WPA	WiFi Protected Access
WPAN	Wireless personal area network
WUSB	Wireless USB
WUWB	Wireless Ultra-Wideband
WWAN	Wireless Wide Area Network
WWW	World Wide Web
ZKDSG	Zugangskontrolldienstschutzgesetz

15 Anhang

15.1 Konfiguration der Entwicklungsumgebung

Die verwendete Entwicklungsumgebung hierbei war Visual Studio 2002 Version 7.0.9466. Dies beinhaltet Auch Microsofts .NET Framework 1.0. Die Umgebung wurde wie folgt konfiguriert:

Unter *Tools* → *Options*→ *Projects*→ *VC++ Directories*→ *Show directories for:* → wurden die folgenden Pfade zu den "included files" hinzugefügt:
C:\openssl-0.9.7c\crypto
C:\openssl-0.9.7c\ssl
C:\openssl-0.9.7c\inc32\openssl
und es wurde ein Pfad für Bibliotheken hinzugefügt: C:\openssl-0.9.7c\out32dll

Unter *project properties* → *Configuration Properties* → *C/C++* → *General* → *Compile As Managed* wird auf "Not using managed extensions" gesetzt.
Anmerkung: Dies wird benötigt, um error C2692 zu beheben

Unter *project properties* → *Configuration Properties* → *C/C++* → *Advanced* → *compile As* wird auf "Compile as C Code (\TC)" gesetzt.

Unter *project properties* → *Configuration Properties* →*Linker* → *Input* → *Additional Dependencies* warden "ssleay32.lib", "libeay32.lib" und "Ws2_32.lib" hinzugefügt.

Damit ist die Entwicklungsumgebung des Prototyps festgelegt.

15.2 Administration von Zertifikaten mit OpenSSL

Abbildung 170 zeigt ein Beispiel von CAs und Komponenten, welche von diesen CAs ausgestellten Zertifikate besitzen passend zu den weiter oben im Hauptteil vorgestellten Roaming Modellen.

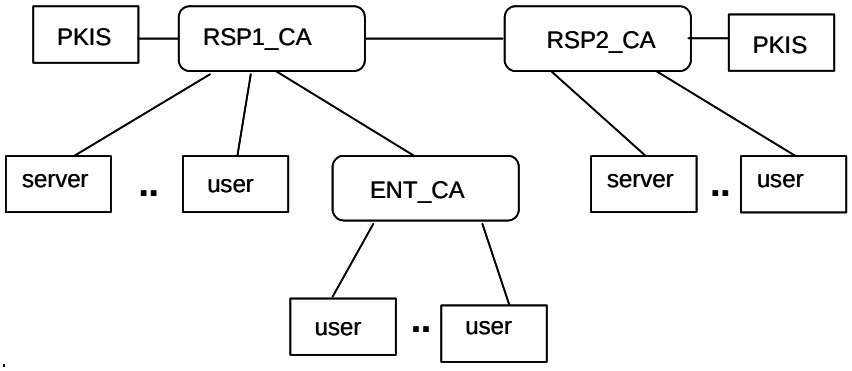


Abbildung 170: CA Hierarchie, wie sie für die Tests in dieser Arbeit eingesetzt wurde

Jede CA stellt ein selbstsigniertes Zertifikat für sich aus. Zusätzlich gilt:

- CAofRSP1 stellt CA-, Server- und Nutzer-Zertifikate aus,
- CAofRSP2 stellt CA-, Server- und Nutzer-Zertifikate aus,
- CAofEnterprise stellt nur Nutzer-Zertifikate aus.

In openssl benötigt jede CA die folgenden Verzeichnisse:

- "certs" enthält alle Zertifikate dieser CA,
- "crl" enthält alle zurückgerufenen Zertifikate,
- "private" enthält die selbstsignierten Zertifikate and die CA-Schlüssel und
- "newcerts" enthält neue Zertifikate.

Die Konfigurationsdatei *openssl.cnf* enthält Informationen darüber, wie die Zertifikate auszustellen sind. Jeder Typ von Zertifikaten wird durch einen speziellen Abschnitt der Datei *openssl.cnf* repräsentiert. Die folgenden Befehle des openssl command-line tools können dabei verwendet werden. Die Parameter sind dabei nur Beispiele.

Zum Ausstellen von Zertifikaten:

- Schlüsselgenerierung:
  >openssl genrsa –des3 –out CAofEnterprise/private/CAkey.pem –rand file1 1024
- to Ausstellen eines selbstsignierten Zertifikates:
  >openssl req –new –x509 –days 365 –key CAofEnterprise/private/CAkey.pem –out CAofEnterprise/private/CAcert.pem –config openssl.cnf
- Kopieren des CA Zertifikates:
  >cp CAofEnterprise/private/CAcert.pem CAofEnterprise/certs/00.pem

- Erzeugung eines Request zur Ausstellung eines Zertifikates:
  >openssl req –new –key CAofRSP2/private/CAkey.pem
  –out CAofRSP2/newcerts/ RSP2CAreq.pem
- Zum Signieren eines Requests zum Ausstellen von Zertifikaten:
  >openssl ca –name CAofRSP1 –keyfile CAofRSP1/private/CAkey.pem
  -in CAofRSP2/newcerts/RSP2CAreq.pem
  -out CAofRSP1/newcerts/RSP2CAcert.pem –outdir CAofRSP1/certs

Abgesehen von den oben gezeigten openssl Befehlen, beinhaltet die Administration von Zertifikaten auch die Konvertierung in verschiedene Formate:

- Schlüssel vom *.pem ins *.key Format konvertieren:
  >openssl rsa –in userkey.pem –out userkey.key
- Zertifikat vom *.pem ins *.crt Format konvertieren:
  >openssl x509 –in usercert.pem –outform DER -out usercert.crt
- a certificate from .pem to pkcs12: >openssl pkcs12 –export –in usercert.pem –inkey userkey.pem –out userfile.p12
- Eine crl vom *.pem ins DER Format konvertieren:
  >openssl crl –in CAofEnterprise/crl/crl.pem –outform DER
  -out CAofEnterprise/crl/crl.crl

Die Sperre von Zertifikaten wird ebenso benötigt:

- Sperrung eines Zertifikates:
  >openssl ca –revoke usercert.pem –name CAofEnterprise
- Erzeugung einer CRL: >openssl ca -gencrl -out CAofEnterprise/crl/crl.pem

15.3 Screenshots Prototyp 1

15.3.1 Not-Extended-Mode

Im nicht erweiterten Modus wird zuerst nach dem Passwort für die Zertifikate gefragt. Danach überprüft der Server, ob Der Client in Modus “0” ist. Wenn dies nicht der Fall ist, dann wird der Handshake abgebrochen. Andernfalls wird der Handshake fortgeführt mit den weiteren Handhsake Nachrichten. Die folgende Abbildung 171 und Abbildung 172 zeigen je eine Instanz eines TLS-Clients und eines TLS Servers nach einem erfolgreichen Handshake.

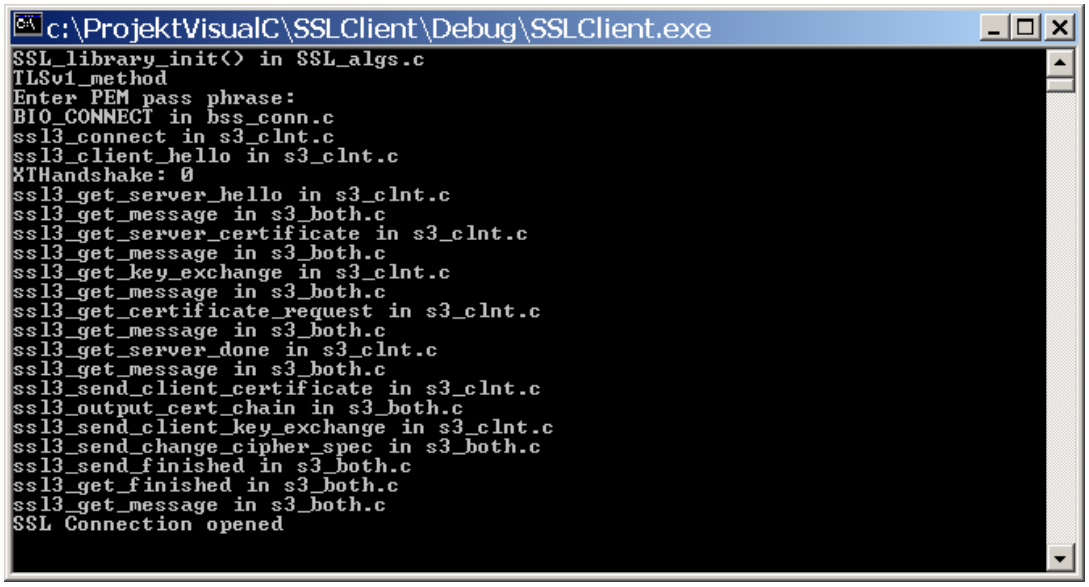


Abbildung 171: TLS-Client nach erfolgreichem Handshake (nicht erweitert)

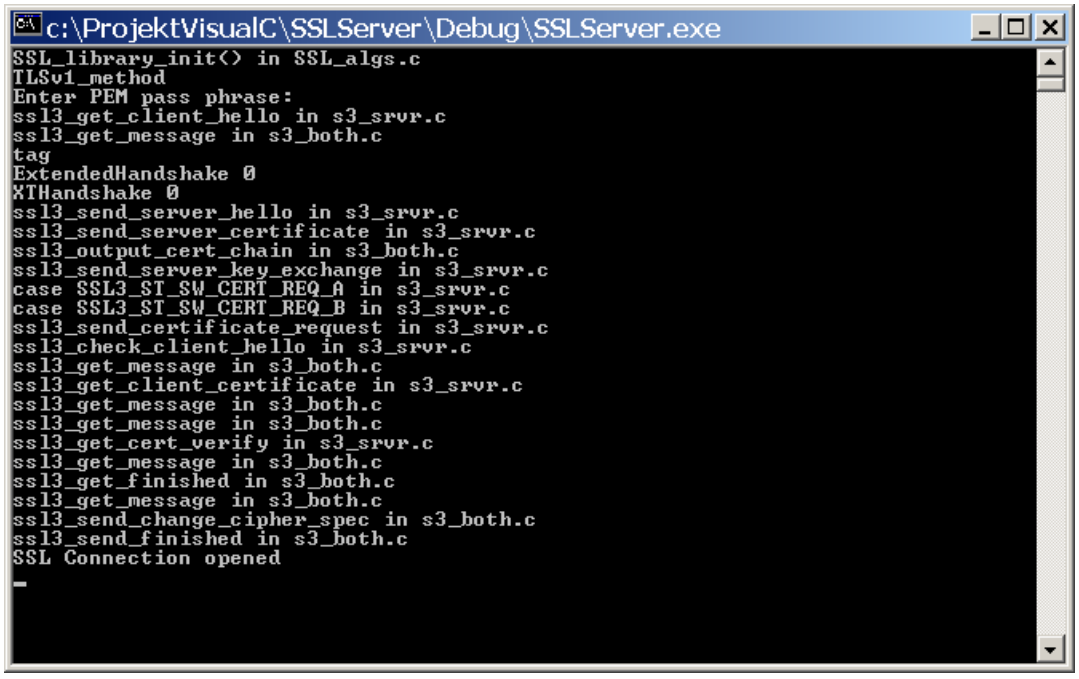


Abbildung 172: TLS Server nach erfolgreichem Handshake (nicht erweitert)

15.3.2 Extended Mode

Im “extended mode” werden bei Client und Server ebenfalls zunächst nach den Passwörtern for ihre jeweiligen Zertifikate gefragt.
Danach Fragt der Client nach Adresse / Hostname des PKI Servers, welcher für die Verifikation des Zertifikates des TLS(SSL) Servers erforderlich ist.

Als nächstes fragt der Server nach der notwendigen Information, um den PKI Server aufzurufen:

Dies ist zum einen der Name der Datei, welche aufgerufen werden muss, zum zweiten die Konfigurationsdatei des PKI-Clients, zum dritten das Zertifikat, welches verwendet werden kann, um die Urheberschaft des PKI Servers zu überprüfen, und zum vierten die Konfigurationsdatei für den Logger des PKI-Client.

Der Client fragt jetzt nach dem Namen der Datei, welche aufgerufen wird, um die PKI-Client Bibliothek zur Verifikation der Signatur des PKI Servers einzusetzen. Als zweites fragt der Client nach der Konfigurationsdatei des PKI-Clients und als drittes nach dem Zertifikat, welches benutzt wird, um die Signatur des PKI Servers zu überprüfen.

Der Server fragt nun nach der notwendigen Information, um den PKI Server, dem er selbst vertraut, zu kontaktieren. Dies ist zuerst der Name der Datei, die aufgerufen werden muss, zweitens die Konfigurationsdatei des PKI Client und drittens das Zertifikat, welches verwendet werden kann, um die Urheberschaft des PKI Servers zu überprüfen und viertens die Konfigurationsdatei für den Logger des PKI-Client.

Als Adresse für den PKI Server, dem der SSL(TLS) Server vertraut ist localhost vorgesehen, d.h. hard-codiert im Programmcode angegeben. Dies kann direkt im Code verändert werden, wenn eine andre Adresse gewünscht ist. Programmcode, der es erlaubt die Adresse manuell einzugeben ist vorhanden aber auskommentiert.
Die Abbildung 173 und die Abbildung 174 zeigen einen TLS-Client und –Server nach einem erfolgreich geschlossenen handshake

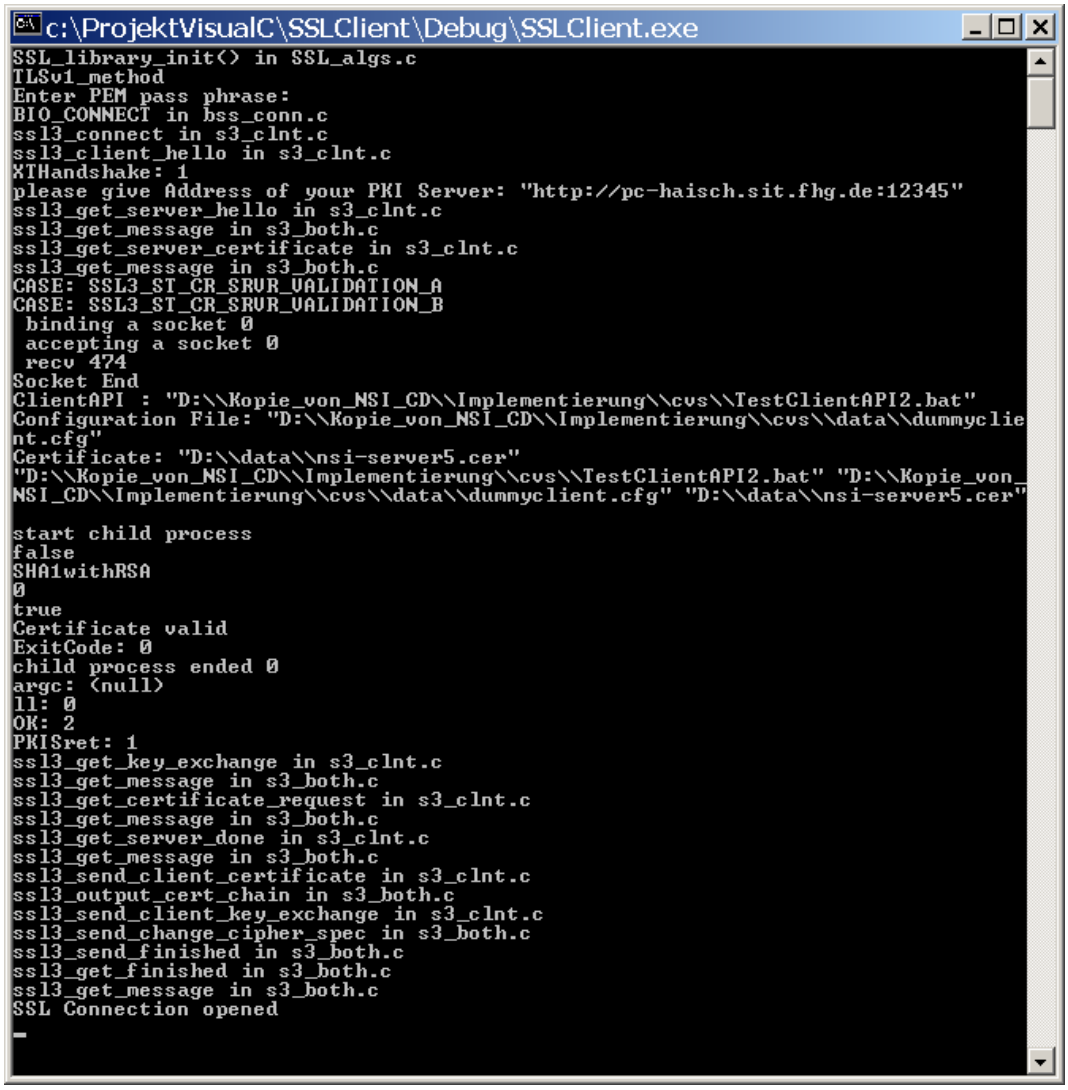


Abbildung 173: TLS-Client nach erfolgreichem Handshake (erweitert)

Abbildung 174: TLS Server nach erfolgreichem Handshake (erweitert)

15.3.3 SSL and TLS

Die vier oben gezeigten Abbildungen zeigen bereits die TLS-Clients und TLS–Server. Um hier in diesem Dokument zu demonstrieren, dass der SSLv3-Handshake ebenfalls durchgeführt werden kann, zeigen die Abbildung 175, Abbildung 176, Abbildung 177 und Abbildung 178 je eine Instanz eines erweiterten und eines nicht erweiterten SSL-Client und Servers nach einem erfolgreich durchgeführten Handshake. Da sie nahezu identisch sind mit den Abbildungen, welche die TLS-Clients und –Server zeigen, werden sie hier nicht weiter erklärt.

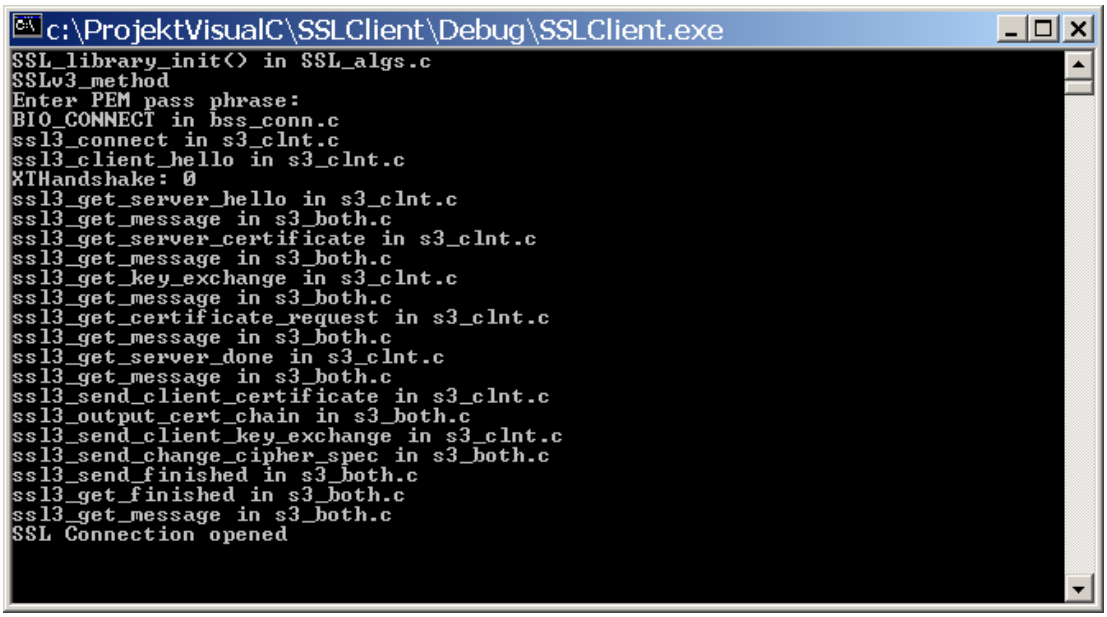


Abbildung 175: SSL-Client nach erfolgreichem Handshake (nicht erweitert)

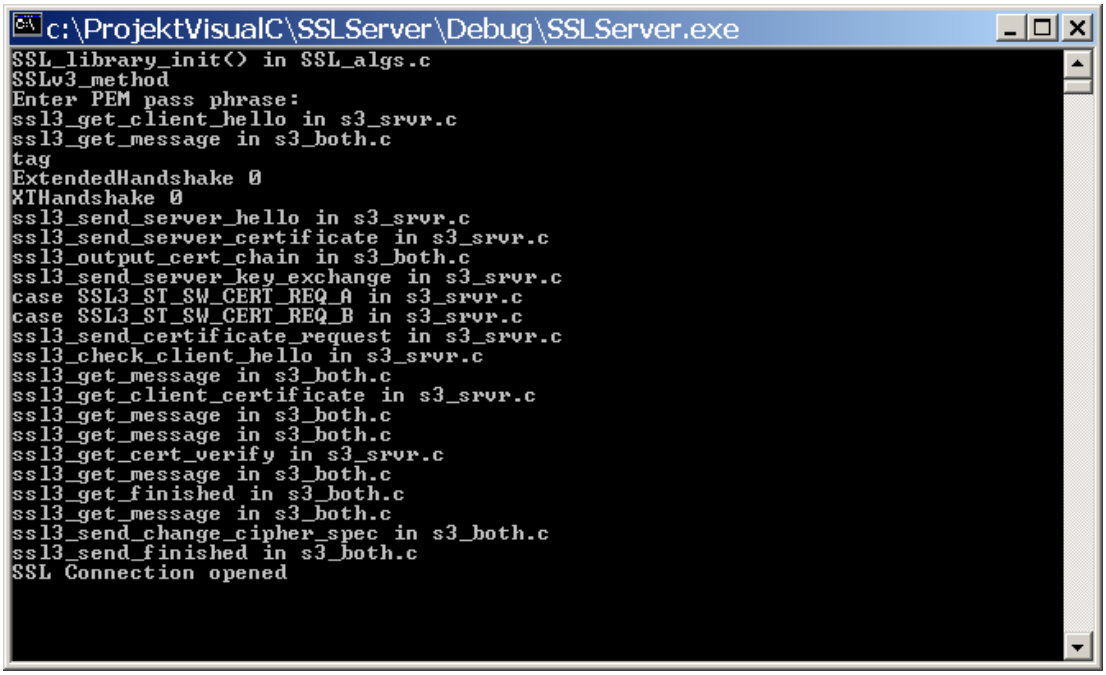


Abbildung 176: SSL Server nach erfolgreichem Handshake (nicht erweitert)

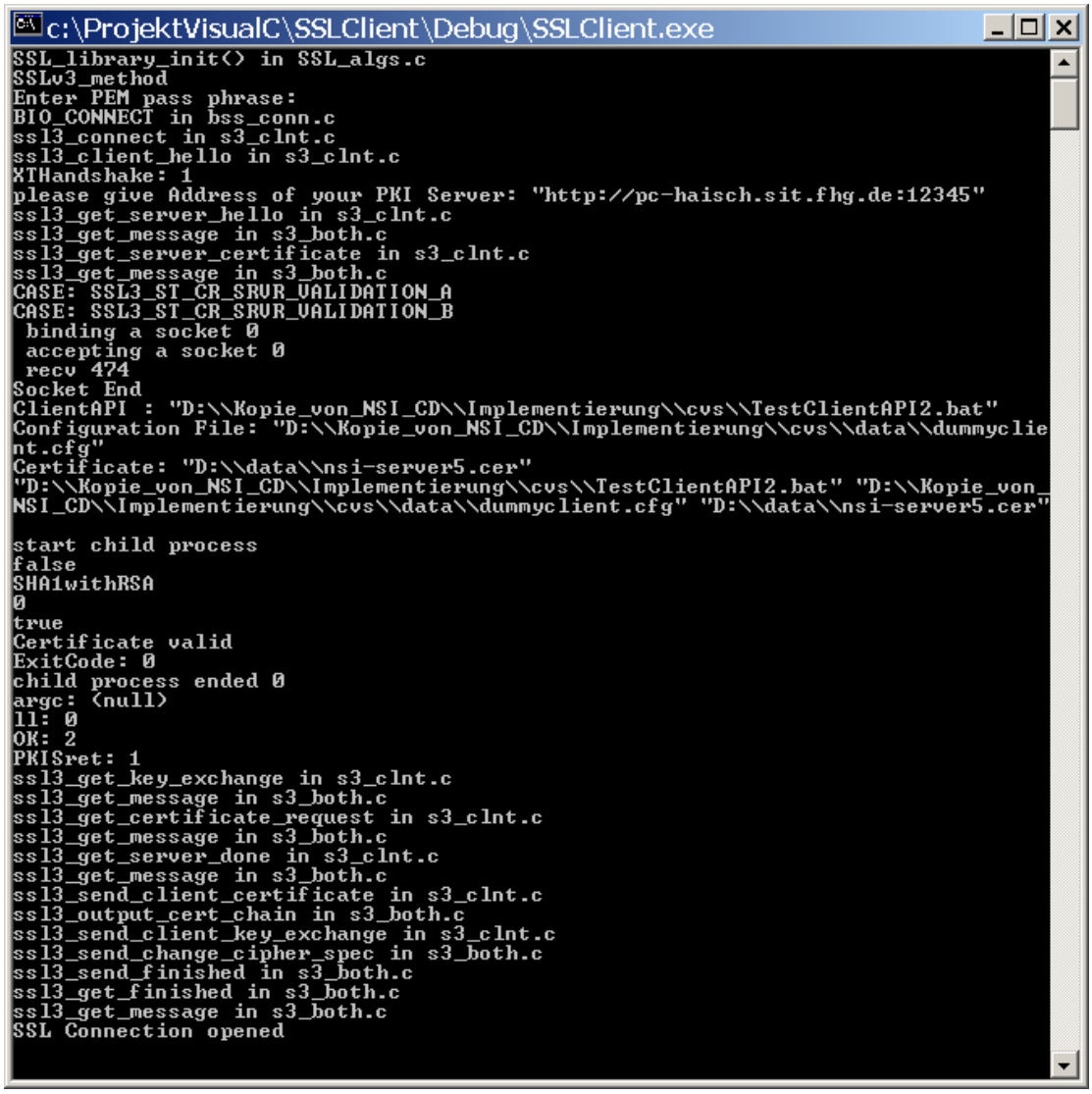


Abbildung 177: SSL-Client nach erfolgreichem Handshake (erweitert)

Abbildung 178: SSL Server nach erfolgreichem Handshake (erweitert)

15.4 Quelldateien zum ersten SSL/TLS-Prototyp

Der Quellcode zum in dieser Arbeit entstandenen ersten Prototyp befindet sich auf der beiliegenden CD innerhalb der Datei Prototyp1.zip. Diese Datei enthält die folgenden fünf Ordner bzw. Verzeichnisse:

- **SSL**: Der Ordner SSL enthält alle modifizierten SSL-Verzeichnisse von OpenSSL. Die Modifikationen sind mit dem Tag "new added code" innerhalb der Dateien markiert. Wenn nur eine einzelne Zeile hinzugefügt ist, dann befindet sich am Ende der Zeile der Kommentar "// new added code". Wenn mehrere Zeilen eingefügt wurden, dann werden sie von "// new added code" Kommentaren eingeschlossen. Gelöschte Zeilen sind auskommentiert und mit dem Hinweis „// deleted code" als Kommentar am Ende der Zeile versehen.
- **ClientServer**: Der Ordner „ClientServer" enthält den SSL(TLS)-Client und Server. Im Ordner certs befindet sich ein Satz Zertifikate für den Betrieb des Client und Servers.
- **Java**: Das Verzeichnis „Java" enthält die *.Java und die zugehörigen *.class Dateien sowie auch die *.bat Dateien, welche zum Aufruf der Java-Programme verwendet werden. Die Java-Programme werden während des Ablaufs des SSL-Handshakes als „Child"-Prozesse aufgerufen. Sie werden verwendet, um mit dem PKIServer zu kommunizieren oder um die PKIClient-Bibliothek zu verwenden.
- **PKIS**: Das Verzeichnis „PKIS" enthält die für den Prototyp verwendete Version des PKIServers und der verwendeten PKIClient Bibliotheken sowie die verwendeten Konfigurationsdateien.
- **CA**: Der Ordner CAs enthält die CAs, welche für das in dieser Arbeit relevante Roaming Szenario erstellt wurden.

15.5 Setup für WLAN/WiMAX Prototyp

15.5.1 Zertifikate

Als erstes muss eine Verzeichnisstruktur angelegt werden, in der die erstellten Zertifikate und Schlüssel abgelegt werden sollen.

Code: /bin/bash

```
1.  mkdir /myCA
2.  mkdir /myCA/certs
3.  mkdir /myCA/private
4.
```

OpenSSL verwendet bei der Erstellung der Zertifikate eine Konfigurationsdatei, in der die verlangten Angaben zum Zertifikatinhaber, Defaultwerte, etc festgelegt sind. Um eine solche Datei zu erstellen ist die einfachste Möglichkeit die vorhandene Beispielkonfiguration zu kopieren und zu modifizieren. Unter Gentoo findet sich diese unter

Code: /bin/bash

```
5.   cp /etc/ssl/openssl.cnf /myCA/myopenssl.cnf
6.
```

Des Weiteren muss noch der Zähler für die Seriennummern initialisiert werden

Code: /bin/bash

```
7.   echo "100001" >> /myCA/serial
8.
```

Für unser Szenario ist es nötig drei PKIs aufzubauen, eine für den NSI Server und jeweils eine weitere für die Zertifikate von Client und Server. Entsprechend muss die Vorbereitungsphase wiederholt werden.

15.5.1.1 NSI-Zertifikate

15.5.1.1.1 NSI-RootCA

Der Anfang einer jeden PKI wird von einem Root-Zertifikat gebildet, das selbstsigniert ist, und von den Nutzern der PKI als Vertrauensanker akzeptiert werden muss. Um ein solches Root-Zertifikat zu erstellen, ist folgender OpenSSL Befehl von nötig

Code: /bin/bash

```
9.   cd [NSI-CA-Pfad]
10.  openssl req -new -x509 -extensions v3_ca -out NSI-Root_Cert.pem
  -keyout
11.        private/NSI-Rootkey.pem -config myopensslnsi.cnf -days 3700
12.
```

15.5.1.1.2 NSI Server Zertifikat

Als nächstes wird ein Zertifikat für den NSI Server erstellt. Die Erstellung von Zertifikaten folgt immer den gleichen Regeln:

1. Es wird ein Request erstellt
2. Der Request wird mit Hilfe des Root-Zertifikats (bzw. dem zugehörigen private key) signiert und ist damit als Zertifikat einsetzbar.
3. Optional kann das Zertifikat zusammen mit dem privaten Schlüssel in einem PKCS#12 Software Token verpackt und an den Besitzer übermittelt werden.

Diese drei Schritte sehen für die Erstellung des NSI Server Zertifikats folgendermaßen aus:

Code: /bin/bash

```
13.  #Request
14.  openssl req -new -nodes -out NSI- Server_Req.pem -keyout
15.        private/NSI- Serverkey.pem -config myopensslnsi.cnf
16.
17.  #Sign
18.  openssl ca -out NSI- Server_Cert.pem -config myopensslnsi.cnf -
  infiles
```

```
19.        NSI- Server_Req.pem
20.
21.  #PKCS#12
22.  openssl pkcsl2 -export -in NSI- Server_Cert.pem -inkey
23.        private/NSI- Server_Key.pem -certfile NSI-Root_Cert.pem
24.        -name "NSI- Server" -out NSI- Server_Cert.pl2 -noiter -
  nomaciter
25.
```

Wenn wir das Software Token auf diese Weise erstellen, haben wir leider das Problem, dass der PKIScout dieses nicht lesen kann, da die Zertifikate Bags vor den Key Bags sind. Diese Ordnung scheint sich mit OpenSSL nicht umkehren zu lassen. Eine Lösung ist das Zertifikat in Mozilla bzw. Firefox zu importieren und anschließend wieder zu exportieren. Die Implementierung von Mozilla speichert die Zertifikate und Key Bags in der Reihenfolge, wie sie vom PKIScout erwünscht sind.

15.5.1.2 Server Zertifikate

Dieses Zertifikat soll später für den Radius Server verwendet werden, der den verbindenden Client authentifiziert.

15.5.1.2.1 (Server) Root CA

Dafür müssen wir zunächst wieder ein selbst signiertes Root Zertifikat erstellen

Code: /bin/bash

```
26.  cd [Server-CA-Pfad]
27.  openssl req -new -x509 -extensions v3_ca -out Server-
  RootCert.pem
28.        -keyout private/Server-Rootkey.pem -config
  myopensslserver.cnf
29.        -days 3700
```

15.5.1.2.2 Server-Zertifikat

Analog zur Erstellung des NSI Server Zertifikats wird auch hier zunächst ein Request erstellt und anschließend signiert. Ein PKCS#12 Software Token müssen wir hier nicht unbedingt erstellen, da die Zertifikate und Schlüssel in der Radius Konfiguration einzeln angegeben werden können.

Code: /bin/bash

```
30.  #Request
31.  openssl req -new -nodes -out ServerReq.pem -keyout
32.        private/ServerKey.pem -config myopensslserver.cnf
33.
34.  #Sign
35.  openssl ca -out ServerCert.pem -config myopensslserver.cnf
36.        -infiles ServerReq.pem
37.
```

Um später das Zertifikat in die Datenbank des PKIScout importieren zu können, wird es in binärer Form benötigt. Die Textform (pem) können wir mittels OpenSSL folgendermaßen in binäre Form (der) umwandeln

Code: /bin/bash

```
38.  openssl x509 -in ServerCert.pem -out ServerCert.der -outform DER
39.
```

15.5.1.2.3 CRLs

Der PKIScout unterstützt die Verwendung von Sperrlisten (CRLs), um zurückgezogene Zertifikate zu erkennen. Mit OpenSSL können wir auf folgende Weise eine CRL für eine CA erstellen:

Code: /bin/bash

```
40.
41.  #generate CRL
42.  openssl ca -gencrl -out crl.pem -config myopensslserver.cnf
43.
44.  #export in DER format
45.  openssl crl -in crl.pem -out crl.der -outform DER
46.
```

Der zweite Befehl ist nötig, da der PKIScout beim Import der CRL ebenfalls nur das binäre Format unterstützt.

15.5.1.3 Client Zertifikate

Um das Testszenario möglich umfassend zu halten, ist es sinnvoll, das Clientzertifikat nicht von der Root-CA des Server signieren zu lassen, sondern noch eine dritte Root-CA zu erstellen. Natürlich muss dafür wieder eine neue Konfigurationsdatei erstellt werden (siehe Vorbereitungen).

15.5.1.3.1 (Client) Root-CA

Völlig analog zum obigen

Code: /bin/bash

```
47.  cd   [Client-CA-Pfad]
48.  openssl req -new -x509 -extensions v3_ca -out Client-
  RootCert.pem
49.        -keyout private/Client-Rootkey.pem -config
  myopensslclient.cnf
50.        -days  3700
```

15.5.1.3.2 Client Zertifikat

Auch hier die üblichen Schritte: Erzeugung des Request und signieren durch die CA:

Code: /bin/bash

```
#Request
openssl req -new -nodes -out ClientReq.pem -keyout
      private/ClientKey.pem -config myopensslclient.cnf

#Sign
openssl ca -out ClientCert.pem -config myopensslclient.cnf
      -infiles ClientReq.pem

```

Abhängig davon, welche Supplikanten Software verwendet wird, kann es nötig sein, ein PKCS#12 Software Token zu erstellen (siehe NSI Server-Zertifikat).

15.5.2 Cisco AP Konfiguration

Im 802.1x Szenario wird ein Authentifikator benötigt. Wird eine drahtlose Infrastruktur betrieben, so ist dies der Access Point. Die folgenden Erläuterungen beziehen sich auf Cisco Access Point der 1200er Serie.

15.5.2.1 Radius Server hinzufügen

Zunächst muss im Access Point der zu verwendende Radius Server in die Liste eingetragen werden. Dazu dient im Web Interface die Seite Security->Server Manager. Dort muss die IP Adresse sowie die verwendeten Ports und das vereinbarte Shared Secret eingetragen werden, wie in Abbildung 179 gezeigt. Anschließend wird die Konfiguration durch Drücken des "apply" Buttons gespeichert.

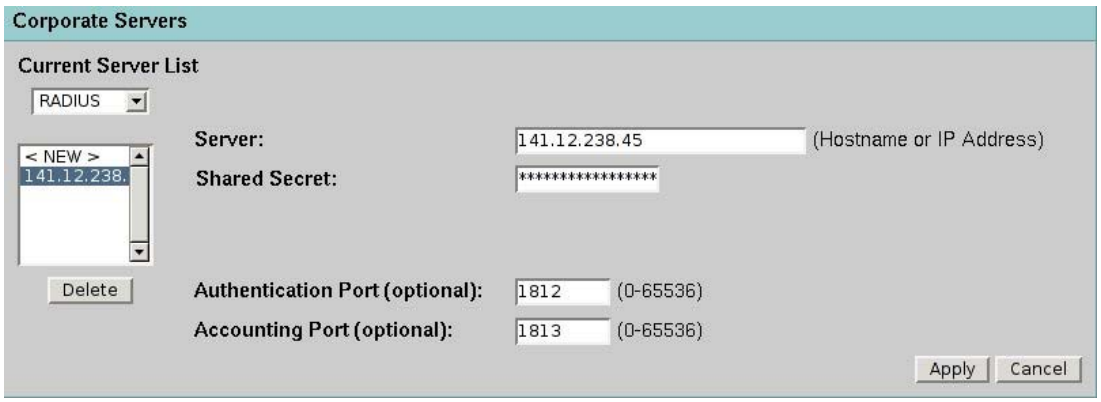


Abbildung 179 Radius Server hinzufügen

Falls gewünscht, kann der neue Radius Server als Default EAP- Server gesetzt werden. Auch hier wird die Auswahl durch Bestätigen mit dem apply Button übernommen. Dies zeigt Abbildung 180.

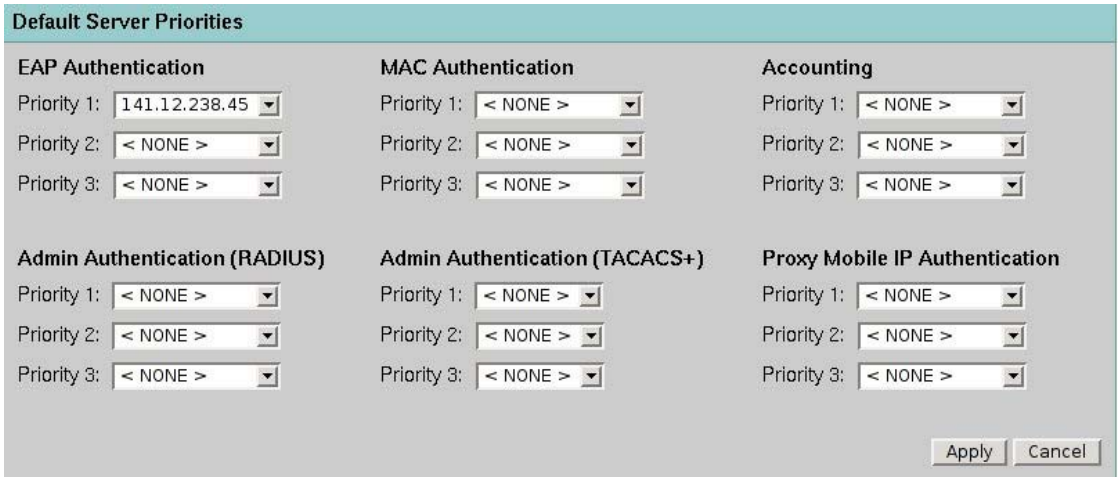


Abbildung 180 Default Server Priorities

15.5.2.2 SSID einrichten

Der nächste Schritt ist das Anlegen einer neuen SSID. Dazu wird der SSID Manager aufgerufen, wie in Abbildung 181 gezeigt. Die nötigen Einstellungen sind die folgenden:

- Ein Name muss festgelegt werden (in unseren Szenario testnet)
- In den Authentication Settings ist unter „Methods Accepted“ zum einen „Open Authentication“ mit der Option „with EAP“ und zum anderen „Network EAP“ ohne Option auszuwählen.
- Falls der konfigurierte Radius Server nicht als „Default“ eingetragen wurde, muss in den „Server Priorities“ auf „Customize“ umgestellt und der Radius Server ausgewählt werden. Ansonsten ist „Use Defaults“ in Ordnung.
- Zuletzt wird durch den apply-Button die Einstellung gespeichert.

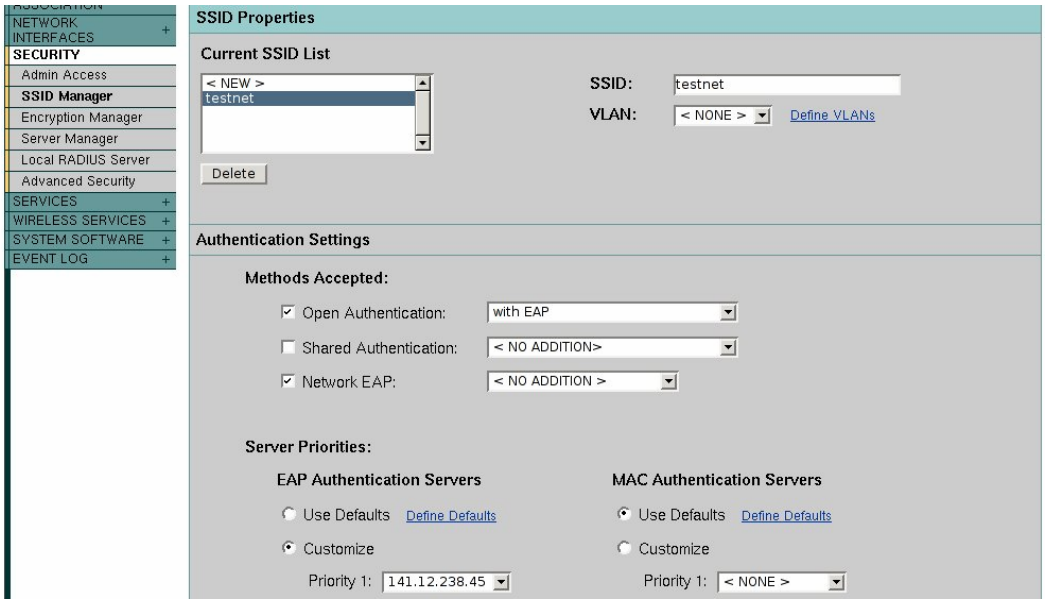


Abbildung 181 SSID einrichten

15.5.2.3 Verschlüsselung

Damit der AP auch eine verschlüsselte Verbindung zum Client aufbaut, muss im „Encryption Manager“ bei den „Encryption Modes“ „WEP“ ausgewählt und auf „Mandatory“ gesetzt werden. Ein WEP-Key muss nicht gesetzt werden, da dieser bei der Authentifikation mit dem Xsupplicanten automatisch erzeugt und erneuert wird. Die Auswahl ist wiederum durch den apply-Button zu bestätigen.

15.5.3 Hostapd

Der hostapd ist eine Software, die es erlaubt einen Linuxrechner als Access Point zu verwenden. In unserem Fall benötigen wir allerdings nur die Funktionalität von hostapd als **Authentifikator** in einem 802.1x Szenario aufzutreten.

15.5.3.1 Installation

Im Gentoo Portage findet sich zwar das Paket net-wireless/hostapd, es erfüllt aber in einigen Punkten unsere Wünsche nicht (Trigger für Authentifizierung ist dhcp-request, kein Port-Based-Network-Access). Speziell für unser gefordertes Wired-Szenario gibt es das Programm „Port Authentication Entity“ unter http://sourceforge.net/projects/pae.

15.5.3.2 PAE Kernel Modul

Das PAE Modul benötigt im Kernel eine Unterstützung für „Layer-2-Hooks“. Ein Kernel-Patch findet sich im PAE-Archiv und wird folgendermaßen angewendet:

Code: /bin/bash

```
59.
60.  cd /usr/src/linux
61.  patch -pl < [pae-pfad]/kernel-patches/layer2-hooks/linux.patch
62.  make && make modules_install
63.
```

Anschließend muss der neue Kernel noch installiert werden (d.h. im Bootmanager eingetragen werden). Danach kann das PAE Modul kompiliert werden

Code: /bin/bash

```
64.  cd [pae-pfad]
65.  make
66.
```

15.5.3.3 Hostapd entpacken und patchen

Entweder ist der Sourcecode von hostapd manuell herunterzuladen oder die aktuelle Version im Gentoo Portage zu benutzen

Code: /bin/bash

```
67.  ebuild /usr/portage/net-wireless/hostapd/hostapd-0.4.7-rl.ebuild
  unpack
68.
```

Danach befindet sich der Sourcecode unter
/var/tmp/portage/hostapd-0.4.7-r1/work.
Nun muss noch der hostapd-Patch des PAE angewandt werden

Code: /bin/bash

```
69.  cd /var/tmp/portage/hostapd-0.4.7-rl/work/hostapd-0.4.7
70.  patch -p1 < [pae-pfad]/hostap-patches/hostap-0.3.5
71.
```

Abhängig von der verwendeten Version des hostapd muss vorher das Patch-file manuell etwas bearbeitet werden.

15.5.3.4 Hostapd Basiskonfiguration

Um die gewünschten Features des Hostapd einzustellen, lohnt sich ein Blick in die Datei *defconfig*

Datei: defconfig

```
72.  # Driver interface for wired authenticator
73.  CONFIG_DRIVER_WIRED=y
74.  CFLAGS += -I../pae/modules            # change to reflect local
  setup
75.
76.  # Integrated EAP server
77.  CONFIG_EAP=y
78.
79.  # EAP-TLS for the integrated EAP server
80.  CONFIG_EAP_TLS=y
81.
```

Die CONFIG_DRIVER_WIRED Option muss für unser Szenario auf jeden Fall gesetzt sein. Die Verwendung des integrierten EAP Servers ist optional.

15.5.3.5 Hostapd kompilieren

War das Patchen erfolgreich und sind die gewünschten Features gewählt, so kann nun der hostapd kompiliert werden

Code: /bin/bash

```
82.  ./configure
83.  make
84.
```

15.5.3.6 Konfiguration

Im Quellcode des hostapd ist eine Beispielkonfiguration für den Fall eines Wired Szenario vorhanden. Dieses kann mit kleinen Veränderungen verwendet werden:

Datei: wired.conf

```
85.  ##### hostapd configuration file
  ######################################## ######
86.  interface=ethl driver=wired
87.  bridge=brO
88.  ieee8021x=l
89.  #eap_reauth_period=3600
90.  use_pae_group_addr=l
91.
92.  ##### RADIUS configuration
  ############################################### ######
93.  for IEEE 802.IX with external Authentication Server, IEEE 802.11
94.  authentication with external ACL for MAC addresses, and
  accounting
95.  The own IP address of the access point (used as NAS-IP-Address)
  ownipaddr=141.12.238.47
96.
97.  RADIUS authentication server
  auth_server_addr=141.12.238.45
  auth_server_port=1812
98.    auth server shared secret=123testl23
99.
```

Jetzt sollte der HostAP einsatzbereit sein und Anfragen von Client entgegennehmen.

15.5.4 Bridge

Um verschiedene Netze zu verbinden, werden Bridges eingesetzt. Die bridge-utils erlauben es, einen Linuxrechner als 802.1d Ethernet-Bridge einzusetzen. Wir benutzen die Bridge Utils, um im Host AP-Rechner die Datenpakete von authentifizierten Client ins interne Netz weiterzuleiten.

15.5.4.1 Kernel Anpassung

Der Kernel muss für die Unterstützung von 802.1d Bridging folgendermaßen konfiguriert werden:

Code: kernel-2.6

```
100. Device Drivers
101.       --> Networking Support
102.             --> Networking Options
103.                   --> <*> 802.1d Ethernet Bridging
104.
```

Die Verwendung als Modul ist ebenfalls möglich.

Code: kernel-2.6 (Module)

```
105. Device Drivers
106.       --> Networking Support
107.             --> Networking Options
108.                   --> <M> 802.1d Ethernet Bridging
109.
```

Dann muss allerdings noch dafür gesorgt werden, dass das Modul auch geladen wird

Datei /etc/modules.autoload/kernel-2.6

```
110. # /etc/modules.autoload.d/kernel-2.6:  kernel modules to load
  when system boots.
111. #
112. # Note that this file is for 2.6 kernels.
113. #
114. # Add the names of modules that you'd like to load when the
  system
115. # starts into this file, one per line.  Comments begin with #
  and
116. # are ignored.  Read man modules.autoload for additional
  details.
117.
118. # For example:
119. # 3c59x
120.
121. bridge
122.
```

15.5.4.2 bridge-utils Installieren

Das Paket zur Verwaltung des Bridging heißt bridge-utils. Es lässt sich unter Gentoo mittels

```
123. emerge bridge-utils
124.
```

installieren. Es stellt den Befehlt **brctl** zur Verfügung.

15.5.4.3 Eine Bridge erstellen

Um mehrere Interfaces an eine bridge zu binden, muss zunächst eine bridge erstellt werden, zu der dann die einzelnen Interfaces hinzugefügt werden. Des Weiteren müssen die Interfaces "up" sein, jedoch darf ihnen keine IP-Adresse zugewiesen sein

Code: /bin/bash

```
125. ifconfig eth0 0.0.0.0 up
126. ifconfig eth1 0.0.0.0 up
127. brctl addbr br0
128. brctl addif br0 eth0
129. brctl addif br0 eth1
130.
131. #set forward delay
132. brctl setfd br0 0
133.
134. #bring up bridge
135. ifconfig br0 <ip> netmask <mask> up
136. route add default gw <gw>
137.
```

15.5.4.4 Bridge automatisch starten

Zum automatischen Start der Bridge dienen die Init Scripts von Gentoo. Zunächst muss die /etc/conf.d/net entsprechend den Anforderungen geändert werden

Datei: /etc/conf.d/net

```
138. #Bridging #########
139. #depend on net.eth0 an net.eth1 if they need extra configuration
140.
141. depend_br0(){
142. net.eth0 net.eth1
143.
144. #Configure Bridge - "man brctl" for more details
145. brctl_br0=( "setfd 0" "sethello 0" "stp off" )
146.
147. #Add ports to bridge
148. bridge_br0="eth0 eth1"
149.
150. #Give the bridge an address
151. #config_br0=( "dhcp" )
152. config_br0=( "141.12.238.45 netmask 255.255.255.224" )
153.
154. #Set ports to null
155. config_eth0=( "null" )
156. config_eth1=( "null" )
157.
```

Sollte kein InitScript für eth0, eth1 oder br0 bestehen, kann einfach das net.lo script verwendet werden. Anschließend wird das net.br0 Script dem default runlevel hinzugefügt (um es beim Systemstart automatisch zu starten) und gestartet.

Code: /bin/bash

```
158. ln -sf /etc/init.d/net.lo /etc/init.d/net.br0
159. rc-update add net.br0 default
160. /etc/init.d/net.br0 start
```

15.5.5 Xsupplicant

15.5.5.1 Paket emergen

Im Gentoo Portage befindet sich zur Zeit die Version 1.2.2 des Xsupplicant, markiert als testing. Diese Version kann mit folgendem Befehl installiert werden:

Code: /bin/bash

```
161. ACCEPT_KEYWORDS="~x86" emerge -v xsupplicant
162.
```

Der Nachteil dieser Methode ist, dass auch von sämtlichen Abhängigkeiten die testing-Version installiert wird, was nicht immer gewünscht ist. Alternativ kann in die Datei /etc/portage/package.keywords folgende Zeile eingefügt werden

Datei: /etc/portage/package.keywords

```
163. net-misc/xsupplicant ~x86
164.
```

, was den oben genannten Nachteil nicht hat.

15.5.5.2 libiw.so.27

Tritt beim Starten des Xsupplicanten der Fehler

```
165. error while loading shared libraries: libiw.so.27: cannot open
  shared object file: No such file or directory
166.
```

auf, wird die falsche Version der wireless-tools verwendet. Der Xsupplicant funktioniert, wie es scheint, nur mit der Version 27. Im Gentoo-Portage ist allerdings bereits die Version 28 als stable gekennzeichnet und wird daher benutzt. Zur Lösung muss ein "downgrade" auf die wireless-tools-27 durchgeführt werden. Dazu wird zunächst in der Datei /etc/portage/package.mask die neuere Version maskiert

Datei: /etc/portage/package.mask

```
167. >net-wireless/wireless-tools-27-rl
168.
```

und anschließend werden die wireless-tools neu "emerget"

Code: /bin/bash

```
169. emerge -pv wireless-tools
170.
```

Hier kann geprüft werden, ob die Änderungen erfolgreich waren. Falls die wireless-tools einer zu hohen Version installiert waren, sollte eine Ausgabe ähnlich der Folgenden geliefert werden

```
171. Calculating dependencies ...done!
172. [ebuild    UD] net-wireless/wireless-tools-27-rl [28_prel0] +nls
  183 kB
173.
```

Dabei steht das D für "downgrade". Wenn die Richtigkeit überprüft ist, kann mit einem "emerge" ohne das „pretend-flag“ das Downgrade gestartet werden.

15.5.5.3 Die Konfigurationsdatei xsupplicant.conf

Datei: /etc/xsupplicant.conf

```
174. ### GLOBAL SECTION
175. networklist = all
176. defaultnetname = default
177. destination = Auto
178. logfile = /var/log/xsupplicant.log
179. authperiod = 30
180. heldperiod = 30
181. maxstarts = 10
182. defaultinterface = eth0
```

```
183. ###  NETWORK SECTION
184. testnet
185. {
186.       type = wireless
187.       allowtypes = eap-tls
188.       identity = TestUser
189.       eaptls {
190.             usercert = /myCA/ClientCert.pem
191.             userkey = /myCA/ClientKey.pem
192.             userkeypass = password
193.             rootcert = /myCA/RootCert.pem
194.             chunksize = 1398
195.             randomfile = /dev/urandom
196.       }
197. }
198. default
199. {
200.       type = wired
201.       allowtypes = eap-tls
202.       identity = TestUser
203.       eaptls {
204.             usercert = /myCA/ClientCert.pem
205.             userkey = /myCA/ClientKey.pem
206.             userkeypass = password
207.             rootcert = /myCA/RootCert.pem
208.             chunksize = 1398
209.             randomfile = /dev/urandom
210.       }
211. }
```

15.5.6 FreeRadius

FreeRadius ist eine Open Source Implementierung eines Radius Servers.

15.5.6.1 Installation

Im Gentoo Portage befindet sich FreeRadius in der Version 1.0.5. Diese besitzt keine besonderen Abhängigkeiten und kann einfach mit

Code: /bin/bash

```
212. emerge -v freeradius
213.
```

installiert werden.

15.5.6.2 Konfiguration

Die Konfigurationsdateien befinden sich standardmäßig (unter Gentoo) im Verzeichnis /etc/raddb. Die wichtigsten Dateien werden im Folgenden kurz angesprochen

15.5.6.3 radiusd.conf

In der Datei radiusd.conf werden die nötigen Informationen für den radiusd Prozess bereitgestellt. Für erste Testläufe sollte die Standardkonfiguration ihren Zweck erfüllen.

15.5.6.4 clients.conf

Hier werden die Client angegeben, die Anfragen an den Server stellen dürfen, und das jeweilige Shared Secret

Datei: clients.conf

```
client 141.12.238.48 {
        secret = 123test123
        shortname  = CiscoAP
}
client 141.12.238.45 {
        secret = 123test123
        shortname = wiredAP
        nastype = other
}
```

In unserem Szenario sind das zum einen der Access Point für die Wireless Authentifizierung und der HostAP-Rechner für den „wired“ Fall.

15.5.7 eap.conf

Die Konfiguration der EAP-Authentifizierung ist aus der radiusd.conf in die Datei eap.conf ausgelagert. Hier werden die Einstellungen für EAP getätigt.

Datei: eap.conf

```
        eap {
                default_eap_type = tls
                timer_expire     = 60

                # Supported EAP-types
                md5 {
                }

                # Cisco LEAP
                leap {
                }

                #  Generic Token Card.
                gtc {
                }

                ## EAP-TLS
                tls {
                        private_key_password = mazze77
                # private_key_file =
  ${raddbdir}/certs/FakeServerPriv.pem
                private_key_file =
  /myCA/second/private/Server_Key.pem

                        #  If Private key & Certificate are located in
                        #  the same file, then private_key_file &
                        #  certificate_file must contain the same file
                        #  name.
                # certificate_file =
  ${raddbdir}/certs/FakeServerCert.pem
                certificate_file = /myCA/second/Server_Cert.pem

```

```
252.                    #  Trusted Root CA list
253.                    CA_path = ${raddbdir}/certs/demoCA
254.                    #CA_file =
  ${raddbdir}/certs/demoCA/FakeRootCA.pem
255.                    #CA_file = /myCA/second/Root_Cert.pem
256.
257.                    dh_file = ${raddbdir}/certs/dh
258.
259.                    random_file = /dev/urandom
260.
261.             }
262.
263.             #  The TTLS module implements the EAP-TTLS protocol,
264.             #  which can be described as EAP inside of Diameter,
265.             #  inside of TLS, inside of EAP, inside of RADIUS...
266.             #ttls {
267.             #}
268.
269.              peap {
270.                    default_eap_type = mschapv2
271.             }
272.
273.             mschapv2 {
274.             }
275.       }
276.
```

Für unser Szenario ist nur EAP-TLS nötig, daher könnten alle anderen Typen auch auskommentiert werden. In den Einstellungen zu TLS werden die benötigten Zertifikate, der Private Key und das zugehörige Passwort angegeben.

15.5.8 PKIScout

Der PKIScout ist die Java-Implementierung eines PKI Servers zur zentralisierten Verifikation von Nutzerzertifikaten. Zur Datenhaltung wird eine MySQL-Datenbank verwendet.

15.5.8.1 Java Installation

Um den PKIScout zu verwenden, wird JAVA benötigt. Unter Gentoo gibt es die Möglichkeit das SUN JDK/JRE oder blackdown JDK/JRE zu installieren. Welches bevorzugt wird, ist jedem selbst überlassen. Im Folgenden wird die Installation des „sun-jre“ beschrieben.

15.5.8.1.1 jre emergen

Code: /bin/bash

```
277. emerge -v sun-jre-bin
278.
```

Hier wird ihnen ein Link zum Download des Softwarepakets präsentiert. Man folgt den „download“-Anweisungen, sprich download nach /usr/portage/distfiles, und führt „emerge“ erneut ausführen. Dieses Mal sollte das jre auch tatsächlich installiert werden.

15.5.8.1.2 konfigurieren

Am Ende der Installation werden sie auf die Notwendigkeit die Umgebungsvariablen upzudaten hingewiesen. Das wird durch folgenden Befehl erledigt

Code: /bin/bash

```
279. env-update && source /etc/profile
280.
```

15.5.8.1.3 spätere Änderungen

Sollten in Zukunft Änderungen an der Konfiguration von JAVA nötig sein, funktioniert das über das Tool **java-config**

15.5.9 MySQL Installation

15.5.9.1 MySQL emergen

Die Optionen für die Installation von MySQL lassen sich folgendermaßen anzeigen:

Code: /bin/bash

```
281. emerge -pv mysql
282.
```

Nachdem die gewünschten Optionen ausgewählt und in /etc/make.conf oder /etc/portage/package.use eingetragen sind, kann der Installationsprozess gestartet werden. Genauere Informationen zu den USE Flags und deren Aufgaben finden sich in diversen Internetresourcen.

15.5.9.2 MySQL Basiskonfiguration

Code: /bin/bash

```
283. #Datenbank initialisieren mysqlinstalldb
284. #MySQL starten
285. /etc/init.d/mysql start
286.
287. #(mysql)root Passwort setzen
288. mysqladmin -u root -h localhost password R00TPASSW0RT
289.
290. #mysql automatisch beim Booten starten
291. rc-update add mysql default
```

15.5.9.3 MySQL Einrichten

15.5.9.3.1 Datenbank "NSI" erstellen

Auf der NSI-CD findet sich ein dumpfile der Tables der Datenbank NSI, die vom PKIScout genutzt wird. Diese Datenbank kann in MySQL mit folgendem Befehl erstellt und gefüllt werden:

Code: /bin/bash

```
292. mysql -h localhost -u root -p NSI < NSIdb.sql
293.
```

Jetzt existiert eine Datenbank mit den benötigten Tables.

15.5.9.3.2 NSI User erstellen

Um einen neuen MySQL Benutzer zu definieren, wird zunächst das mysql Command Line Interface gestartet

Code: /bin/bash

```
294. mysql -p -h localhost
295.
```

Anschließend werden dem neuen Benutzer Rechte zugeteilt, und er wird dadurch gleichzeitig erstellt

Code: /bin/bash

```
296. SET PASSWORD FOR NSI@localhost=PASSWORD('nsipassword');
297. grant CREATE,INSERT,SELECT,DELETE,UPDATE on NSI.* to
  NSI@localhost
298. grant CREATE,INSERT,SELECT,DELETE,UPDATE on NSI.* to NSI;
299. exit
```

15.5.9.3.3 Policy erstellen

Um den PKIScout zu verwenden, benötigen wir noch Strategien, mittels derer entschieden wird, ob ein Zertifikat den Status „gültig“ erhält, oder nicht. Eine solche Strategie wird folgendermaßen mit dem admin.Cli des PKIScout erstellt:

Code: /bin/bash

```
300. java -cp pkiServer-full.jar:FlexiProvider-1.1.5pl.signed.jar
301.       de.fhg.nsi.server.admin.Cli policy install -f
302.       /myCA/Server-RootCert.der -a 01.01.2007 -D -C 1.2.3.4  VAL
303.
```

(Genauere Informationen zur Verwendung der adminCli finden sich in der zugehörigen Dokumentation) Dieser Befehl installiert das Root-Zertifikat des Servers als Vertrauensanker der „default“-Strategie mit Revokations-(Sperr-)Mechanismus CRL.

15.5.9.3.4 CRL importieren

Da wir in der Strategie die Verwendung von CRLs gefordert haben, müssen wir auch aktuelle CRLs angeben, um eine erfolgreiche Prüfung zu erhalten:

Code: /bin/bash

```
304. java -cp pkiServer-full.jar:FlexiProvider-1.1.5pl.signed.jar
```

```
305.       de.fhg.nsi.server.admin.Cli crl imp -f /myCA/Server-
  Rootcrl.der
306.
```

15.5.9.3.5 Funktionstest

Nun sind wir in der Lage, ein Zertifikat mit Hilfe des PKIScout auf Gültigkeit zu prüfen.

Code: /bin/bash

```
307. java -cp pkiServer-full.jar:FlexiProvider-1.1.5pl.signed.jar
308.       de.fhg.nsi.server.admin.Cli policy validateCert -f
309.       /myCA/Server-Cert.der -D
310.
```

Die Ausgabe sollte in etwa folgendermaßen aussehen:

Code: Ausgabe

```
311. Validation Result: VALID
312. Validation Time: 24.08.2005
313. Used Policy : 1.2.3.4
```

15.5.10 TestShell des Servers

15.5.10.1 Aufgabe

Die TestShell erstellt die Verbindung zum PKIScout, setzt die Parameter und startet die Verifikation. Das vom PKI Server zurückgegeben Ergebnis wird anschließend in der Datei PKISRESULT.tmp gespeichert.

15.5.10.2 Aufruf

Das testClientAPI ShellScript sorgt für den Aufruf der TestShell und übergibt die benötigten Parameter.

Datei: testClientAPI

```
314. #!/bin/sh
315. if [ $# -ne 6 ] ;then
316.       echo "Usage : ./testclientAPI <configfile> <cert-to-be-
  tested>  <clientlogger> <prot://host:port> <policy>
  <VerifySignature>"
317.       echo $@
318. exit fi
319.
320. NSI_CD=/NSI_CD
321. CVS=$NSI_CD/Implementierung/cvs
322. LIB=$CVS/lib
323. #PFX=$CVS/build/lib
324. CLS=$CVS/build/classes
325. buildPath ()
326. {
327.       result="";
```

```
328.        mypfx=$l;
329.        files="ls $mypfx/*.jar";
330.        for I in $files; do
331.        result="$I:$result"; done; echo $result;
332. }
333.
334. jars="_buildPath $LIB":$CLS
335. Java -cp $jars de . fhg.nsi.clientapi.TestShell $@
```

15.5.10.3 Der Quell Code

Datei: TestShell.java

```
package de.fhg.nsi.clientapi;

import de.flexiprovider.core.FlexiCoreProvider;
import de.fhg.nsi.clientapi.*;
import de.fhg.nsi.clientapi.searchpath.*;
import de.fhg.nsi.clientapi.valcert.*;
import de.fhg.nsi.asn1.dpvdpd.*;
import de.fhg.nsi.clientapi.policy.PolicyID;
import codec.asn1.*;
import codec.*;
import codec.x509.X509Certificate;
import org.apache.log4j.Logger;
import org.apache.log4j.Level;
import org.apache.log4j.PropertyConfigurator;
import java.security.Provider;
import junit.framework.Test;
import junit.framework.TestCase;
import junit.framework.TestSuite;
import java.util.Collection;
import java.util.Iterator;
import java.io.*;
import java.security.cert.CertificateException;
import java.security.cert.CertificateFactory;
import java.security.cert.Certificate;
import java.net.URL;
import java.io.*;
import de.fhg.nsi.clientapi.logging.*;

public class TestShell {

  public static void main (String[] Args) {

        Logger mLogger;
        Context mContext;
        Core mCore;
        SecurityContext mSecurityContext;
        PkiServerInfo mPkiServerInfo;

        X509Certificate cert;
        CertOrCertRef certorcertref;
        File mFile,mFile1,mFile2;

        byte[] result;
        String status;
        int l1, l2;
        URL ServerAdress;
        String SrvrAdrStr;
        boolean Vfyssgntr;
        String policy;
```

```
Vfyssgntr = false;
if ( Args[5].equals("TRUE") ) Vfyssgntr = true;
System.out.println(Vfyssgntr);
policy = "1.2.3.9";
policy = Args[4];


PropertyConfigurator.configure(Args[2]);
mLogger = Logger.getLogger(TestShell.class);
try {
        FlexiCoreProvider flexiProvider_ = new FlexiCoreProvider();
        java.security.Security.addProvider(flexiProvider_);

        // Instantiate the client-api with the configuration file as parameter
        mCore = new Core(Args[0]);

        // Create new ServerInfo for new server
        mPkiServerInfo = new PkiServerInfo();

        // Set ServerAddress
        SrvrAdrStr = Args[3];
        ServerAdress = new URL(SrvrAdrStr);
        mPkiServerInfo.pkiSetServerURL(ServerAdress);

        // get and set certs to authenticate the server and the servers signature
        mFile1 = new File("/myCA/second/NSI Server_Cert.der");
        Certificate ServerAuthCert = new X509Certificate(new FileInputStream(mFile1));
        mFile2 = new File("/myCA/second/NSI Server_Cert.der");
        Certificate ServerSigCert = new X509Certificate(new FileInputStream(mFile2));
        mPkiServerInfo.pkiSetServerAuthCert(ServerAuthCert);
        mPkiServerInfo.pkiSetServerSigCert(ServerSigCert);

        // set the Validation policy to be used
        mPkiServerInfo.pkiSetPolicyID(new PolicyID( policy, 0));
        // set the Signature policy to be used
        mPkiServerInfo.pkiSetSigPolicyID(new PolicyID("1.2.3.5", 0));

        // set new security context with all possible configuration parameters
        mSecurityContext = new SecurityContext(   null,  //Certificate pClientAuthCert,
                                                  null,  //Certificate pClientSigCert,
                        "/myCA/second/NSI Server_Cert_repaired.p12",  //String psPseUrl,
                                                  false, //boolean pbClientAuthentication,
                                                  false, //boolean pbClientSigning,
                                                  false, //boolean pbServerAuthentication,
                                                  Vfyssgntr, //boolean pbServerSigning,
                                                  false, //boolean pbEncryptConnection,
                                                  true,  //boolean pbReplayProtection,
                                          "1.3.14.3.2.26"); //String psHashAlgorithm)

        // set security context for server
        mPkiServerInfo.pkiSetSecurityContext(mSecurityContext);

        // set connection time out
        mPkiServerInfo.pkiSetConnectionTimeOut(9000000);

        // create new context and add new server to the created context
        mContext = new Context();
        mContext.pkiAddServer(mPkiServerInfo);

        mFile = new File(Args[1]);
        cert = new X509Certificate(new FileInputStream(mFile));
        certorcertref = new CertOrCertRef(cert);
        System.out.println("Sending  request...");

        // call PKIServer to validate the certificate
        mCore.pkiValidateCert(mContext,certorcertref,null);
```

```
                    // answer of PKIServer
                    StatusCode sc = mContext.pkiGetReturnedStatusCode();
                    System.out.println("Connection to PKIServer : "+sc.pkiGetMajorStatusCodeText());

                    if (sc.pkiGetMajorStatusCodeText().equals("Failure"))
                            System.exit(-1);
                    System.out.println("Certificate : "+sc.pkiGetMinorStatusCodeText());

                    status = sc.pkiGetMinorStatusCodeText();
                    l1 = status.length();
                    result = mContext.pkiGetServerResponse();
                    l2 = result.length;

                    System.out.println(l1);          System.out.println(l2);          System.out.println(status);
                    //System.out.println(result);

                    FileOutputStream fos = new FileOutputStream ("PKISRESULT.tmp");
                    DataOutputStream dos = new DataOutputStream (fos);
                    dos.writeInt(l1); // 4Bytes
                    dos.writeInt(l2); // 4Bytes
                    dos.writeUTF(status);
                    dos.write(result);
                    dos.close(); //close Stream
    }
    catch (Exception e) {
      e.printStackTrace();
    }

  }
}
```

15.5.11 TestShell des Clients

15.5.11.1 Aufgabe

Die Aufgabe der TestShell2 auf der Clientseite ist es, das vom Server empfangene PKISRESULT zu analysieren. Dazu wird zunächst getestet, ob die Signatur des PKIServers gültig ist, um anschließend das Ergebnis des PKIServers auszulesen und mit einem entsprechenden Rückgabewert zu beenden.

15.5.11.2 Aufruf

Die TestShell2 wird analog zur Serverseite über das ShellScript testClientAPI aufgerufen

Datei: testClientAPI

```
336. #!/bin/sh
337. #if [ $# -ne 6 ];then
338. echo "Usage : ./testClientAPI <configfile> <cert-to-be-tested>
  <clientlogger> <prot://host:port> <policy> <VerifySignature>"
339. echo $@
340. exit
  #fi
341. NSI_CD=/NSI_CD
342. #CVS=$NSI_CD/Implementierung/cvs
343. CVS=/backup/client/myPKIScout
344. LIB=$CVS/libs #PFX=$CVS/build/lib CLS=$CVS/classes
345. buildPath ()
346. {
347. result="";
348. mypfx=$1;
```

```
349. files="ls $mypfx/*.jar";
350. for I in $files; do
351. result="$I:$result"; done; echo $result;
352. }
353. jars="_buildPath $LIB":$CLS
354. Java -cp $jars:/backup/client/myPKIScout/TestShell2.class
355. de.fhg.nsi.clientapi.TestShell2 $@
```

15.5.11.3 Der Code

Datei: TestShell2.java

```
package de.fhg.nsi.clientapi;

import de.flexiprovider.core.FlexiCoreProvider;
import de.fhg.nsi.clientapi.*;
import de.fhg.nsi.clientapi.searchpath.*;
import de.fhg.nsi.clientapi.valcert.*;
import de.fhg.nsi.asn1.dpvdpd.*;
import de.fhg.nsi.clientapi.policy.PolicyID;
import codec.asn1.*;
import codec.*;
import codec.x509.X509Certificate;
import org.apache.log4j.Logger;
import org.apache.log4j.Level;
import org.apache.log4j.PropertyConfigurator;
import java.security.Provider;
import junit.framework.Test;
import junit.framework.TestCase;
import junit.framework.TestSuite;
import java.util.Collection;
import java.util.Iterator;
import java.io.*;
import java.security.cert.CertificateException;
import java.security.cert.CertificateFactory;
import java.security.cert.Certificate;
import java.net.URL;
import java.io.*;
import de.fhg.nsi.clientapi.logging.*;
import java.security.cert.*;

import de.fhg.nsi.asn1.common.ASN1Signature;
import codec.x509.AlgorithmIdentifier;
import de.fhg.nsi.clientapi.Core;

public class TestShell2 {

        public static void main (String[] Args) {

        Context mContext;
        Core mCore;
        X509Certificate cert;
        CertOrCertRef certorcertref;

        File mFile2;

        String status;
        int l1,l2;

        ASN1DPVResponse response;
        ASN1Signature signature;

   try {
```

```
            Logger mLogger = Logger.getLogger(TestShell2.class);////////////
            FlexiCoreProvider flexiProvider_ = new FlexiCoreProvider();
            java.security.Security.addProvider(flexiProvider_);

            // Instantiate the client-api with the configuration file as parameter
            mCore = new Core(Args[0]);

            FileInputStream fis = new FileInputStream ("PKISRESULT_NEU.tmp");
            DataInputStream dis = new DataInputStream (fis);
            l1 = dis.readInt();
            l2 = dis.readInt();
            String erg = dis.readUTF();
            byte[] result = new byte[l2];
            dis.read(result);
            dis.close();

            response = new ASN1DPVResponse(result);

            boolean SignatureExistant = response.getOptionalSignature().isOptional();
                        // true nicht da die signatur
            System.out.println(SignatureExistant);
            if (!SignatureExistant) {
                        mFile2 = new File(Args[1]);
                        cert = new X509Certificate(new FileInputStream(mFile2));
                        java.security.cert.X509Certificate servercert = (java.security.cert.X509Certificate)  cert;
//pPkiServerInfo.pkiGetServerSigCert();

                        signature = response.getOptionalSignature();
                        byte[] sig = signature.exportSignature();

                        AlgorithmIdentifier AlgID = signature.getSignatureAlgorithm();
                        String sAlg = AlgID.getAlgorithmName(); // toString
                        System.out.println(sAlg);

                        ASN1TbsDPVResponseData ToBeSignedData = response.getTbsResponseData();
                        byte[] tbsMess = ToBeSignedData.getEncoded();

                        ASN1SingleDPVResponses DPVresponses = ToBeSignedData.getCertsProcessed();
                        ASN1SingleDPVResponseData SiResDat = (de.fhg.nsi.asn1.dpvdpd.ASN1SingleDPVResponseData)
DPVresponses.get(0);
                        ASN1ValidationStatus ASN1ValStat = SiResDat.getValidationStatus();

                        int statusINT = ASN1ValStat.getChoice();
                        System.out.println(statusINT);

                        boolean SignatureValid =
                                    Core.cryptoVerifySignature(tbsMess, sig, sAlg, servercert);

                        System.out.println(SignatureValid);
                        System.out.println(erg);

                        if ( SignatureValid && ( ( statusINT == 1 && erg.equals("Certificate invalid") ) || (statusINT == 2 &&
erg.equals("Certificate unknown")) || ( statusINT == 0 && erg.equals("Certificate valid") ) ) ) System.exit(statusINT);
//System.out.println(statusINT);
                        else System.exit(4);

            };
  }
  catch (Exception e) { e.printStackTrace();
                        System.exit(4)          ;
  }
  }
}
```

15.5.12 OpenSSL

15.5.12.1 Gentoo Portage

15.5.12.1.1 Was ist Portage

Sämtliche Applikationen sind unter Gentoo im sog. Portage abgelegt. Dort befinden sich

- ebuild-files verschiedener Programmversionen und
- diverse Patches für die einzelnen Programme und Versionen

gespeichert.

In den ebuild-files ist ein Pfad zum Quellcode des jeweiligen Programms angegeben, sowie diverse Funktionen, die bei der Installation ausgeführt werden sollen, unter anderem auch welche Patches angewendet werden sollen. Der Portagebaum wird durch ein

```
356. emerge   --sync
357.
```

mit dem aktuellen Portage der Gentoo-Entwickler synchronisiert.

15.5.13 Warum Portage Overlay

Die Möglichkeit Patches zu verwenden, erlaubt es jedem einzelnen Benutzer, Programme nach seinen Wünschen zu verändern, indem er ein neues Patchfile erzeugt und im ebuild des Programms die Anweisung zur Anwendung des Patches gibt. Leider werden jedoch sämtliche Änderungen beim nächsten „emerge sync“ wieder überschrieben. Genau hier setzt das Portage-Overlay ein. Es erlaubt dem Benutzer ein eigenes Portage Verzeichnis anzugeben, das bei der Synchronisation nicht überschrieben wird und transparent über das normale Portage gelegt wird.

15.5.14 Portage Overlay erstellen

Zunächst muss ein Verzeichnis erstellt werden, das als Portage Overlay verwendet werden soll.

Code: /bin/bash

```
358. mkdir /usr/local/portage
359.
```

Dann muss emerge mitgeteilt werden, dass ein Portage Overlay existiert und wo es zu finden ist.

Datei: /etc/make.conf

```
360. PORTDIR OVERLAY="/usr/local/bin"
361.
```

15.5.15 Einen ebuild-file modifizieren

Im Folgenden wird die Verwendung des Portage-Overlay am Beispiel einer Modifikation von OpenSSL dargestellt.

Das Portage Overlay muss exakt die gleiche Struktur haben wie das normale Portage, d.h. es müssen im Falle von OpenSSL folgende Verzeichnisse erstellt werden:

Code: /bin/bash

```
362. mkdir /usr/local/portage/dev-libs
363. mkdir /usr/local/portage/dev-libs/openssl
364.
```

Anschließend wird das zu verändernde ebuild-file und die darin verwendeten Patches aus dem orginalen Portage in das neue Verzeichnis kopiert.

Code: /bin/bash

```
365. cp /usr/portage/dev-libs/openssl/openssl-0.9.7e-rl.ebuild
366.       /usr/local/portage/dev-libs/openssl/
367.
368. cp /usr/portage/dev-libs/openssl/files
369.       /usr/local/portage/dev-libs/openssl/
370.
```

Zusätzlich muss noch das neue PatchFile kopiert werden. Nun kann in einem Editor das ebuild geöffnet werden, und in der Funktion src_unpack() unser Patch hinzugefügt werden. Angenommen der Dateiname des Patches lautet openssl-0.9.7e-mypatch.patch und liegt im Verzeichnis /usr/local/portage/dev-libs/openssl/files

Datei: /usr/local/portage/dev-libs/openssl/openssl-0.9.7e-rl.ebuild

```
371. src_unpack()
372. {
373.       unpack  ${A}
374.       epatch "${FILESDIR}M/${PN}-0.9.7g-superh.patch
375.       epatch "${FILESDIR}"/${PN}-0.9.7g-amd64-fbsd.patch
376.       epatch "${FILESDIR}M/${PN}-0.9.7e-mypatch.patch
377.       ...
378.
```

Nach der Modifikation des ebuild muss der Digest neu erstellt werden, da sonst emerge beim Überprüfen der Checksum einen Fehler produziert

Code: /bin/bash

```
379. ebuild /usr/local/portage/dev-libs/openssl/openssl-0.9.7e-
  rl.ebuild
380.       digest
381.
```

Jetzt kann das veränderte OpenSSL aus dem lokalen Portage Overlay installiert werden

Code: /bin/bash

```
382. emerge -pv openssl
383.
384. These are the  packages  that  I would  merge,   in  order:
385. Calculating  dependencies   ...done!
386. [ebuild       R       ]   dev-libs/openssl-0.9.7e-rl     -
  bindist   -emacs   -test  0  kB   [1]
387. Total   size  of  downloads:   0  kB Portage  overlays:
388. [1]   /usr/local/portage
389.
```

An dieser Ausgabe kann überprüft werden, ob emerge tatsächlich das Overlay benutzt. An der [1] hinter dem openssl und der Fußnote erkennen wir, dass alles richtig ist.

15.5.15.1 Client Modifikation

15.5.15.1.1 Erweiterung des SSL Kontext Objekts

Das SSL-Context Objekt wird um einen booleschen Wert XTHandshake erweitert, der signalisiert, ob der Extended Handshake verwendet wird. Weiterhin wird die Länge des übergebenen Strings, der die Adresse des PKI Servers angibt, als SERVERIDSIZE definiert und im SSL-Context ein String dieser Länge reserviert. Letzteres ist nötig, da sonst bei jeder Reauthentifikation (falls diese Option im Authentifikator aktiviert ist) erneut nach dem PKI Server gefragt wird, was nicht erwünscht ist.

Code: Definitionen und Erweiterungen

```
390. --- ssl/ssl.h     2005-09-11 12:46:07.000000000 +0200
391. +++ ssl/ssl.h     2005-10-31 13:07:28.000000000 +0100
392. @@ -190,6 +190,7 @@
393.   */
394.  #define SSL_SESSION_ASN1_VERSION 0x0001
395.
396. +#define    SERVER_ID_SIZE                16    // XTHSK
397.  /* text strings for the ciphers */
398.  #define SSL_TXT_NULL_WITH_MD5           SSL2_TXT_NULL_WITH_MD5

399.  #define SSL_TXT_RC4_128_WITH_MD5   SSL2_TXT_RC4_128_WITH_MD5

400. @@ -929,6 +930,8 @@
401.       int first_packet;
402.       int client_version;     /* what was passed, used for
403.                           * SSLv3/TLS rollback check */
404. +     unsigned char XTHSK_server[SERVER_ID_SIZE];     // XTHSK
405. +     int XTHandshake;  // XTHSK
406.       };
407.
408.  #ifdef __cplusplus
```

15.5.15.1.2 Extended Handshake Aktivierung

Es wird dem Client die Möglichkeit überlassen, einen "normalen" SSL Handshake durchzuführen oder den neuen "extended Handshake". Diese Auswahl wird über eine Umgebungsvariable namens XTHSK geregelt. Ist sie auf 1 gesetzt wird der erweiterte Handshake durchgeführt.

Code: XTHSK Umgebungsvariable auslesen

```
409. --- ssl/ssl_lib.c 2005-09-11 12:46:07.000000000 +0200 +++
  ssl/ssllib.c 2005-10-31 14:39:27.000000000 +0100 @@ -295,6 +295,17
  @@
410. s->references=1;
411. s->server=(ctx->method->ssl_accept ==
  sslundefined_function)?0:1;
412. +     strncpy(s->XTHSK_server,"",SERVERIDSIZE);;
413. +
414. +     char* envstring = getenv("XTHSK");  //check if XTHSK
415. enabled
416. +     if (envstring != NULL) {
417. +     int xthsk = atoi(envstring);
418. +     if (xthsk == 1)
419. +     s->XTHandshake = 1;
420. +     eise
421. +     s->XTHandshake = 0;
422. SSLclear(s);
423. CRYPTO_new_ex_data(CRYPTO_EX_INDEX_SSL, s, &s->ex_data);
```

15.5.15.1.3 Modifikation des Client Hello

Zunächst müssen wir im Client Hello dem Server mitteilen, dass ein Extended Handshake durchgeführt werden soll und welcher PKI Server verwendet werden soll. Dazu wird auf der Standardeingabe nach der Adresse des PKI Servers gefragt. Sollte im SSL-Context bereits ein Wert gesetzt sein, wird dieser ohne Nachfrage verwendet, da es sich um eine Reauthentifikation handelt. Im Client Hello Paket wird der Adressstring folgendermaßen abgelegt:

- Eingeleitet durch den Tag XTHSK,
- gefolgt von der Länge des Strings als Integer und
- der String selbst

Code: Erweitertes Client Hello

```
424. --- ssl/s3_clnt.c 2005-09-11 12:46:07.000000000 +0200
425. +++ ssl/s3_clnt.c 2005-11-04 08:54:12.000000000 +0100 @@ -549,6
  +562,27 @@
426. memcpy(p,s->s3->client_random,SSL3RANDOMSIZE);
427. p+=SSL3_RANDOM_SIZE;
428. +     /* Extended Handshake (new code) */
429. +     //printdebug(XTHSK,"Initialization: XTHandshake = %d",
  XTHandshake);
430. +     if( s->XTHandshake ){
431. +     unsigned char server_ID[SERVER_ID_SIZE];
432. +     unsigned char tag[5] = "XTHSK";
433. +
434. +     if (strcmp(s->XTHSK_server,"") == 0){
435. +     printdebug(XTHSK,"please give address of your PKI server:
  ");
436. +     scanf("%s", &server_ID);
437. +     strncpy(s->XTHSK_server,server ID,SERVER_ID SIZE);
438. +     }
439. +     else
440. +     strncpy(server_ID,s->XTHSK_server,SERVER_ID SIZE);
441. +     i = strlen(serverlD);
442. +     memcpy(p,tag,5);
```

```
443. +     p+=5;
444.  memcpy(p,serverlD,i) ;  p+=i;
445. /* Session ID */ if (s->new_session) i=0;
```

15.5.15.1.4 Modifikation des Zustandsautomaten

Bei der Verwendung eines PKI Servers sind einige neue Zustandsübergange nötig.

- Nach dem Certificate Request des Servers wird die Antwort des PKIServers erwartet, anstatt einem Server Done
- Der neue Zustand SSL3_ST_CR_PKI_ANSWER muss abgearbeitet werden und bei Erfolg in den Zustand SSL3_ST_CR_SRVR_DONE wechseln

Code: Zustandsautomat

```
446. --- ssl/s3_clnt.c 2005-09-11 12:46:07.000000000 +0200
447. +++ ssl/s3_clnt.c 2005-11-04 08:54:12.000000000 +0100
448. @@ -302,7 +304,10 @@
449.              case SSL3_ST_CR_CERT_REQ_B:
450.                    ret=ssl3_get_certificate_request(s);
451.                    if (ret <= 0) goto end;
452. -                  s->state=SSL3_ST_CR_SRVR_DONE_A;
453. +                  if (s->XTHandshake)     // XTHSK
454. +                        s->state=SSL3_ST_CR_PKI_ANSWER_A;   //
  XTHSK
455. +                  else  // XTHSK
456. +                        s->state=SSL3_ST_CR_SRVR_DONE_A;    //
  XTHSK
457.                    s->init_num=0;
458.                    break;
459.
460. @@ -318,6 +323,14 @@
461.
462.                    break;
463.
464. +            case SSL3_ST_CR_PKI_ANSWER_A: // start XTHSK
465. +            case SSL3_ST_CR_PKI_ANSWER_B:
466. +                  ret=ssl3_get_pki_answer(s);
467. +                  if (ret <= 0) goto end;
468. +                  s->state=SSL3_ST_CR_SRVR_DONE_A;
469. +                  s->init_num=0;
470. +                  break;      // end XTHSK
471. +
472.              case SSL3_ST_CW_CERT_A:
473.              case SSL3_ST_CW_CERT_B:
474.              case SSL3_ST_CW_CERT_C:
```

15.5.15.1.5 Deaktivierung der Standardverifikation

Da der Client sich nur auf das Ergebnis des PKIServers verlässt, muss die normale Zertifikatsprüfung deaktiviert werden.

Code: Deaktivierte Zertifikatsprüfung

```
475. --- ssl/s3_clnt.c 2005-09-11 12:46:07.000000000 +0200 +++
  ssl/s3_clnt.c 2005-11-04 08:54:12.000000000 +0100 @@ -832,7 +866,9
  @@
```

```
476. p=q;
477. i=ssl_verifycertchain(s,sk);
478. +      // i=ssl_verifycertchain(s,sk);  //Mazzemod: Do NOT
  verify server cert here
479. if ((s->verify_mode != SSLVERIFYNONE) && (!i) #ifndef
  0PENSSLN0KRB5
480. && (s->s3->tmp.new_cipher->algorithms & (SSL_MKEY_MASK| SSL AUTH
  MASK))
```

15.5.15.1.6 Verarbeitung des PKIS-Result

Die neue Funktion

static int ssl3_get_pki_answer(SSL *s)

ist für die Überprüfung der vom Server gelieferten PKI-Antwort vorgesehen.

Code: ssl3_get_pki_answer

```
481. --- ssl/s3_clnt.c 2005-09-11 12:46:07.000000000 +0200
482. +++ ssl/s3_clnt.c 2005-11-04 08:54:12.000000000 +0100
483. @@ -118,6 +118,7 @@
484. #include <openssl/evp.h>
485. #include <openssl/md5.h>
486. #include <openssl/fips.h>
487. +#include <sys/wait.h> //added for WEXITSTATUS
488. static SSLMETHOD *ssl3_get_client_method(int ver);
489. static int ssl3_client_hello(SSL *s);
490. @@ -131,6 +132,8 @@
491. static int ssl3_get_key_exchange(SSL *s);
492. static int ssl3_get_server_certificate(SSL *s);
493. static int ssl3_check_cert_and_algorithm(SSL *s);
494. +static int ssl3_get_pki_answer(SSL *s);
495. +static void printdebug(int level,char* message);
496. static SSLMETHOD *ssl3_get_client_method(int ver)
497. {
498. if (ver == SSL3 VERSION)
499. @@ -1988,4 +2024,111 @@ err:
500. return(0);
501. }
502. +static int ssl3_get_pki_answer(SSL *s){ +
503. +     int al;
504. +     long n;
505. +     int ok,i;
506. +     unsigned char *p;
507. +     unsigned char pkianswer[255];
508. +     n=ssl3_get_message(s,
  +  SSL3_ST_CR_PKI_ANSWER_A,
509. +     SSL3_ST_CR_PKI_ANSWER_B,
510. +     -1, //SSL3MTPKIANSWER,
511. +     -1, // should be changed to something meaningful
512. +     &ok);
513. +     if (!ok) return((int)n);
  +
514. +     p=(unsigned char *)s->init_msg;
515. +     FILE *fpResult;
516. +     fpResult = fopen("PKISRESULT_NEU.tmp","wb");
517. +     fwrite(p,1,n,fpResult);
518. +     fclose(fpResult);
```

```
519. +     print_debug(XTHSK_Debug,"PKISRESULTNEU.tmp received and
  saved
520. ");
521. +     /*char *api = "/work/myPKIScout/testClientAPI";
522. +     char *client_config =
  "/work/myPKIScout/info/dummyclient.cfg";
523. +     char *pkiserver_cert = "/myCA/second/NSI ServerCert.der";
  +
524. +     int length =
  strlen(api)+strlen(clientconfig)+strlen(pkiserver_
  cert);
  +
525. +     char cmd[1024];   //maximale laenge ?
526. +     sprintf(cmd,"%s %s %s",api,clientconfig,pkiservercert);
527. +     printdebug(XTHSKDebug,cmd);
528. +     */
529. +
530. +     char api[256];
531. +     print_debug(XTHSK,"ClientAPI: \n");
532. +     scanf("%s",api);
533. +
534. +     char clientconfig[256];
535. +     printdebug(XTHSK,"Client Configuration File: \n");
536. +     scanf("%s",clientconfig);
537. +
538. +     char pkiservercert[256];
539. +     print_debug(XTHSK,"PKIServer Certificate: \n");
540. +     scanf("%s",pkiservercert);
541. +
542. +     char cmd[768]; //max length ?
543. +     sprintf(cmd,"%s %s %s",api,clientconfig,pkiservercert);
544. +
545. +     printdebug(XTHSK,"Commandline for PKIServer Signature
546. Verification: ");
547. +     printf("%s\n",cmd);
548. +     fflush(stdout);
549. +     int checksig = system(cmd);
550. +     if(remove("PKISRESULTNEU.tmp")!=0)
551. +     print_debug(XTHSK,"Error removing PKISRESULTNEU.tmp !!!
552. Please delete manually...");
553. +     switch(WEXITSTATUS(checksig)){
554. +     case 127 :
555. +     printdebug(XTHSKDebug,"Signature Validation: error in
  system-call");
556. +     al=SSL3_AD_ERR0R_IN_SYSTEM_CALL;
557. +     SSLerr(SSL_F_SSL3_PKI_SERVER_VERIFY,ERR_R_FATAL);
558. +     goto f_err;
559. +     case 126 :
560. +     printdebug(XTHSKDebug,"Signature Validation: Calling
  ClientAPI : Permission denied");
561. +     al=SSL3_AD_ERR0R_IN_SYSTEM_CALL;
562. +     SSLerr(SSL_F_SSL3_PKI_SERVER_VERIFY,ERR_R_FATAL);
563. +     goto f_err;
564. +
565. +     case 0 :
566. +     printdebug(XTHSK,"Signature Validation: Return value = 0,
  Certificate Valid\n");
567. +     return 1;
568. +     case 1 :
569. +     printdebug(XTHSK,"Signature Validation: Return value = 1,
  Certificate Invalid");
570. +     al=SSL3_AD_CERTIFICATE_UNKN0WN;
```

```
571. +     SSLerr(SSL_F_SSL3_PKI_SERVER_VERIFY,ERR_R_FATAL);
572. +     goto f_err;
573. +     case 2 :
574. +     printdebug(XTHSK,"Signature Validation: return value = 2,
  Certificate Unknown");
575. +     al=SSL3_AD_BAD_CERTIFICATE;
576. +     SSLerr(SSL_F_SSL3_PKI_SERVER_VERIFY,ERR_R_FATAL);
577. +     goto ferr;
578. +
579. +     default:
580. +     printdebug(XTHSKError,"Signature Validation:
581. Singature not valid or problem in decoding PKISRESULTNEU.tmp
  ...");
582. +     al=SSL3_AD_PKISRESULT_ERR0R;
583. +     SSLerr(SSL_F_SSL3_PKI_SERVER_VERIFY,ERR_R_FATAL);
584. +     goto f_err;
585. +
586. +     goto f_err;
587. +     }
588. +     return(-1);
589. +f_err:
590. +     ssl3_send_alert(s,SSL3_AL_FATAL,al);
591. +     return(0);
592. + }
593. +void printdebug(int level,char* message){
594. +     switch(level){
595. +     case XTHSK :
596. +          printf("[XTHSK] %s\n",message);
597. +          break;
598. +     case XTHSK Error :
599. +          printf("[XTHSK-Error] %s\n",message);
600. +          break;
601. +     case XTHSKDebug :
602. +          printf("[XTHSK-Debug] %s\n",message);
603. +          break;
604. +     fflush(stdout);
```

15.5.15.1.7 Benötigte Definitionen

In den obigen Quellcodes wurden einige Bezeichner verwendet, die noch definiert werden müssen. Dazu zählen die Codierungen der Zustände

Code: Zustandscodierung

```
605. --- ssl/ssl3.h    2005-09-11 12:46:07.000000000 +0200
606. +++ ssl/ssl3.h    2005-10-31 11:49:35.000000000 +0100
607. @@ -430,6 +432,8 @@
608.  /* read from server */
609.  #define SSL3_ST_CR_SRVR_HELLO_A          (0x120|SSL_ST_CONNECT)
610.  #define SSL3_ST_CR_SRVR_HELLO_B          (0x121|SSL_ST_CONNECT)
611. +#define SSL3_ST_CR_PKI_ANSWER_A          (0x125|SSL_ST_CONNECT)
          // XTHSK
612. +#define SSL3_ST_CR_PKI_ANSWER_B          (0x126|SSL_ST_CONNECT)
          // XTHSK
613.  #define SSL3_ST_CR_CERT_A          (0x130|SSL_ST_CONNECT)
614.  #define SSL3_ST_CR_CERT_B          (0x131|SSL_ST_CONNECT)
615.  #define SSL3_ST_CR_KEY_EXCH_A            (0x140|SSL_ST_CONNECT)
616.
```

und deren Beschreibung für Debugzwecke

Code: Zustandsbeschreibung

```
617. --- ssl/ssl_stat.c      2005-09-11 12:46:07.000000000 +0200
618. +++ ssl/ssl_stat.c      2005-10-31 11:26:36.000000000 +0100
619. @@ -129,6 +129,9 @@
620.  case SSL3_ST_CR_CERT_REQ_B:  str="SSLv3 read server certificate
  request B"; break;
621.  case SSL3_ST_CR_SRVR_DONE_A: str="SSLv3 read server done A";
  break;
622.  case SSL3_ST_CR_SRVR_DONE_B: str="SSLv3 read server done B";
  break;
623. +case SSL3_ST_CR_PKI_ANSWER_A:      str="SSLv3 read
  PKISRESULT_A"; break;      // XTHSK
624. +case SSL3_ST_CR_PKI_ANSWER_B:      str="SSLv3 read
  PKISRESULT_B"; break;      // XTHSK
625. +
626.  case SSL3_ST_CW_CERT_A:            str="SSLv3 write client
  certificate A"; break;
627.  case SSL3_ST_CW_CERT_B:            str="SSLv3 write client
  certificate B"; break;
628.  case SSL3_ST_CW_CERT_C:            str="SSLv3 write client
  certificate C"; break;
629. @@ -280,6 +283,9 @@
630.  case SSL3_ST_CR_CERT_REQ_B:              str="3RCR_B"; break;
631.  case SSL3_ST_CR_SRVR_DONE_A:             str="3RSD_A"; break;
632.  case SSL3_ST_CR_SRVR_DONE_B:             str="3RSD_B"; break;
633. +case SSL3_ST_CR_PKI_ANSWER_A:            str="3RPA_A"; break;
           // XTHSK
634. +case SSL3_ST_CR_PKI_ANSWER_B:            str="3RPA_B"; break;
           // XTHSK
635. +
636.  case SSL3_ST_CW_CERT_A:                        str="3WCC_A";
  break;
637.  case SSL3_ST_CW_CERT_B:                        str="3WCC_B";
  break;
638.  case SSL3_ST_CW_CERT_C:                        str="3WCC_C";
  break;
639.
```

sowie die Message Types für die PKI Server Antwort

Code: Message Type PKI_RESULT

```
640. --- ssl/ssl3.h    2005-09-11 12:46:07.000000000 +0200
641. +++ ssl/ssl3.h    2005-10-31 11:49:35.000000000 +0100
642. @@ -507,6 +511,12 @@
643.  #define SSL3_MT_CLIENT_KEY_EXCHANGE           16
644.  #define SSL3_MT_FINISHED                20
645.
646. +     // start XTHSK
647. +#define SSL3_MT_PKI_RESULT              4
648. +#define XTHSK          1
649. +#define XTHSK_Debug          2
650. +#define XTHSK_Error          3
651. +     // end XTHSK
652.  #define SSL3_MT_CCS                     1
653.
654.  /* These are used when changing over to a new cipher */
```

Die Definitionen von XTHSK, XTHSKDebug und XTHSKError werden in der Funktion *print_debug(int level,char* message)* als Ausgabetyp verwendet und erlauben, die Menge an Information, die an den Benutzer ausgegeben wird, einzustellen.

15.5.15.1.8 Fehlerbehandlung

Im Folgenden werden noch die zur Fehlerbehandlung benötigten Definitionen angegeben.

Code t1_enc.c

```
655. --- ssl/t1_enc.c  2005-09-11 12:46:07.000000000 +0200
656. +++ ssl/t1_enc.c  2005-10-31 11:21:29.000000000 +0100
657. @@ -816,6 +816,9 @@
658.        case
  SSL_AD_CERTIFICATE_EXPIRED:return(SSL3_AD_CERTIFICATE_EXPIRED);
659.        case
  SSL_AD_CERTIFICATE_UNKNOWN:return(SSL3_AD_CERTIFICATE_UNKNOWN);
660.        case SSL_AD_ILLEGAL_PARAMETER:
     return(SSL3_AD_ILLEGAL_PARAMETER);
661. +      case SSL_AD_ERROR_IN_SYSTEM_CALL:
662.             return(SSL3_AD_ERROR_IN_SYSTEM_CALL);          //
  XTHSK
663. +      case SSL_AD_PKI_SCOUT_CONNECTION_FAILURE:
664.             return(SSL3_AD_PKI_SCOUT_CONNECTION_FAILURE);  //
  XTHSK
665. +      case SSL_AD_PKISRESULT_ERROR:
666.             return(SSL3_AD_PKISRESULT_ERROR);              //
  XTHSK
667.        case SSL_AD_UNKNOWN_CA:       return(TLS1_AD_UNKNOWN_CA);
668.        case SSL_AD_ACCESS_DENIED:
     return(TLS1_AD_ACCESS_DENIED);
669.        case SSL_AD_DECODE_ERROR:     return(TLS1_AD_DECODE_ERROR);
```

Code ssl.h--- ssl/ssl.h 2005-09-11 12:46:07.000000000 +0200

```
670. +++ ssl/ssl.h 2005-10-31 13:07:28.000000000 +0100
671. @@ -1068,6 +1071,9 @@
672.  #define SSL_AD_CERTIFICATE_EXPIRED SSL3_AD_CERTIFICATE_EXPIRED
673.  #define SSL_AD_CERTIFICATE_UNKNOWN SSL3_AD_CERTIFICATE_UNKNOWN
674.  #define SSL_AD_ILLEGAL_PARAMETER   SSL3_AD_ILLEGAL_PARAMETER
  /* fatal */
675. +#define    SSL_AD_ERROR_IN_SYSTEM_CALL
     SSL3_AD_ERROR_IN_SYSTEM_CALL
676.             /*fatal */        // XTHSK
677. +#define    SSL_AD_PKI_SCOUT_CONNECTION_FAILURE
678.             SSL3_AD_PKI_SCOUT_CONNECTION_FAILURE /*fatal*/ //
  XTHSK
679. +#define    SSL_AD_PKISRESULT_ERROR
     SSL3_AD_PKISRESULT_ERROR
680.             /*fatal*/   // XTHSK
681.  #define SSL_AD_UNKNOWN_CA          TLS1_AD_UNKNOWN_CA      /*
  fatal */
682.  #define SSL_AD_ACCESS_DENIED       TLS1_AD_ACCESS_DENIED   /*
  fatal */
683.  #define SSL_AD_DECODE_ERROR        TLS1_AD_DECODE_ERROR    /*
  fatal */
```

Code: ssl3.h

```
684. --- ssl/ssl3.h    2005-09-11 12:46:07.000000000 +0200
685. +++ ssl/ssl3.h    2005-10-31 11:49:35.000000000 +0100
686. @@ -280,7 +280,9 @@
687.  #define SSL3_AD_CERTIFICATE_EXPIRED     45
688.  #define SSL3_AD_CERTIFICATE_UNKNOWN     46
689.  #define SSL3_AD_ILLEGAL_PARAMETER 47    /* fatal */
690. -
691. +#define    SSL3_AD_ERROR_IN_SYSTEM_CALL  61 /*fatal */
     // XTHSK
692. +#define    SSL3_AD_PKI_SCOUT_CONNECTION_FAILURE     62
  /*fatal*/ //XTHSK
693. +#define    SSL3_AD_PKISRESULT_ERROR           63 /*fatal*/
     // XTHSK
694.  typedef struct ssl3_record_st
695.       {
696.  /*r */     int type;               /* type of record */
```

Code: ssl stat.c

```
697. --- ssl/ssl_stat.c      2005-09-11 12:46:07.000000000 +0200
698. +++ ssl/ssl_stat.c      2005-10-31 11:26:36.000000000 +0100
699. @@ -387,6 +393,9 @@
700.       case SSL3_AD_CERTIFICATE_EXPIRED:   str="CE"; break;
701.       case SSL3_AD_CERTIFICATE_UNKNOWN:   str="CU"; break;
702.       case SSL3_AD_ILLEGAL_PARAMETER:          str="IP"; break;
703. +     case SSL3_AD_ERROR_IN_SYSTEM_CALL:  str="SC";break;
     // XTHSK
704. +     case SSL3_AD_PKI_SCOUT_CONNECTION_FAILURE: str="CF"; break;
  // XTHSK
705. +     case SSL3_AD_PKISRESULT_ERROR:           str="PE"; break
     // XTHSK
706.       case TLS1_AD_DECRYPTION_FAILED:          str="DC"; break;
707.       case TLS1_AD_RECORD_OVERFLOW:       str="RO"; break;
708.       case TLS1_AD_UNKNOWN_CA:            str="CA"; break;
709. @@ -446,6 +455,15 @@
710.       case SSL3_AD_ILLEGAL_PARAMETER:
711.            str="illegal parameter";
712.            break;
713. +     case SSL3_AD_ERROR_IN_SYSTEM_CALL:       // start XTHSK
714. +          str="Failed to call ClientAPI";
715. +          break;
716. +     case SSL3_AD_PKI_SCOUT_CONNECTION_FAILURE:     //added
717. +          str="Failed to connect to PKIServer";
718. +          break;
719. +     case SSL3_AD_PKISRESULT_ERROR:      //added
720. +          str="Error opening PKISRESULT.tmp";
721. +          break;                              // end XTHSK
722.       case TLS1_AD_DECRYPTION_FAILED:
723.            str="decryption failed";
724.            break;
```

Code: ssl.h

```
725. --- ssl/ssl.h     2005-09-11 12:46:07.000000000 +0200
726. +++ ssl/ssl.h     2005-10-31 13:07:28.000000000 +0100
727. @@ -1549,6 +1555,9 @@
728.  #define SSL_F_SSL3_SETUP_KEY_BLOCK              157
729.  #define SSL_F_SSL3_WRITE_BYTES                          158
730.  #define SSL_F_SSL3_WRITE_PENDING                159
731. +#define SSL_F_SSL3_PKI_CLIENT_VERIFY                 250
     // XTHSK
732. +#define SSL_F_SSL3_PKI_SERVER_SIGANTURE       251         //
  XTHSK
733. +#define SSL_F_SSL3_PKI_SERVER_VERIFY                 252
     // XTHSK
734.  #define SSL_F_SSL_ADD_DIR_CERT_SUBJECTS_TO_STACK     215
735.  #define SSL_F_SSL_ADD_FILE_CERT_SUBJECTS_TO_STACK    216
736.  #define SSL_F_SSL_BAD_METHOD                    160
737.
```

15.5.15.2 Server Modifikation

15.5.15.2.1 Extended Handshake Verwendung

Um sich zu merken, ob vom Client ein Extended Handshake verlangt wurde wird eine boolesche Variable XTHandshake eingeführt und mit dem Defaultwert false initialisiert.

Code:

```
738. ---  ssl/s3_srvr.c      2005-09-07 10:10:11.000000000 +0200
739. +++  ssl/s3_srvr.c      2005-10-31 15:21:55.000000000 +0100
740. @@ -127,6 +127,8 @@
741.  #include <openssl/md5.h>
742.  #include <openssl/fips.h>
743.
744. +#include <sys/wait.h>        // XTHSK
745. +int XTHandshake = 0;                // XTHSK
746.  static SSL_METHOD *ssl3_get_server_method(int ver);
747.  static int ssl3_get_client_hello(SSL *s);
748.  static int ssl3_check_client_hello(SSL *s);
```

15.5.15.2.2 Extended Handshake Erkennung

Der Client sendet im Client Hello Paket ein Feld XTHSK mit der Adresse des PKI Servers. Der Server muss also in der Verarbeitungsroutine des Client Hello prüfen, ob das XTHSK-Feld existiert

Code: ssl3_get_client_hello

```
749. ---  ssl/s3_srvr.c      2005-09-07 10:10:11.000000000 +0200
750. +++  ssl/s3_srvr.c      2005-10-31 15:21:55.000000000 +0100
751. @@ -696,6 +733,30 @@
752.       memcpy(s->s3->client_random,p,SSL3_RANDOM_SIZE);
753.       p+=SSL3_RANDOM_SIZE;
754.
755. +     /* are we using XTHandshake ? */          // start XTHSK
756. +           unsigned char tag[5];
757. +           memcpy(tag,p,5);
758. +           if (strncmp(tag,"XTHSK",5) == 0){
759. +                 XTHandshake = 1;
```

```
760. +                  printf("[XTHSK] XTHandshake is set => PKIServer
  enabled.\n");
761. +                  p+=5;
762. +                  i=*(p++);
763. +                  if(i > 15)
764. +                  {
765. +                              // TODO
766. +                        //SSLerr(SSL_F_SSL3_GET_CLIENT_HELLO,
  SSL_R_PKI_SERVER_ADDRESS_TOO_LARGE);
767. +                        //al = SSL_AD_PKI_SERVER_ERROR;
768. +                        printf("server address too long !!! \n");
769. +                        goto err;
770. +
771. +                  }
772. +                  memcpy(s->PKI_server,p,i);
773. +                  p+=i;
774. +                  s->PKI_server[i]='\0';
775. +                  printf("[XTHSK] Using PKI server address %s
  !\n",s->PKI_server);
776. +            }
777. +
778. +            // stop XTHSK
779.       /* get the session-id */
780.       j= *(p++);
781.
```

Existiert das Feld, so wird der Wert in eine neu eingeführte Variable des SSL-Context geschrieben. Die Variable PKIServer wurde folgendermaßen eingefügt:

Code: SSLContext Erweiterung

```
782. --- ssl/ssl.h 2005-09-07 10:10:11.000000000 +0200
783. +++ ssl/ssl.h 2005-10-31 15:02:53.000000000 +0100
784. @@ -189,6 +189,7 @@
785. * Version 1 - added the optional peer certificate
786. */
787. #define SSLSESSIONASN1VERSION 0x0001
788. + #define SERVER ID SIZE      16
789. /* text strings for the ciphers */
790. #define SSL_TXT_NULL_WITH_MD5 SSL2_TXT_NULL_WITH_MD5
791. @@ -929,6 +930,7 @@
792. int firstpacket;
793. int clientversion;       /* what was passed, used for
794. * SSLv3/TLS rollback check */
795. +       unsigned char PKIserver[SERVERIDSIZE];
796. #ifdef      cplusplus
```

15.5.15.2.3 Änderungen am Zustandsautomaten

Analog zu den Modifikationen beim Client müssen auch beim Server neue Zustandsübergänge eingeführt werden. So muss bei Verwendung des Extended Handshake vor dem SSL3_ST_SW_SRVR_DONE ein neuer Zustand SSL3_ST_SW_PKI ANSWER eingefügt werden, der die PKI Server Antwort verarbeitet.

Code: Zustandsautomat

```
797. @@ -384,7 +387,10 @@
```

```
798.                          /* no cert request */
799.                          skip=1;
800.                          s->s3->tmp.cert_request=0;
801. -                        s->state=SSL3_ST_SW_SRVR_DONE_A;
802. +                        if(XTHandshake)                    //
  startXTHSK
803. +                              s->state=SSL3_ST_SW_PKI_ANSWER_A;
804. +                        else
805. +                              s->state=SSL3_ST_SW_SRVR_DONE_A;
  //end XTHSK
806.                          }
807.                   else
808.                          {
809. @@ -392,10 +398,17 @@
810.                          ret=ssl3_send_certificate_request(s);
811.                          if (ret <= 0) goto end;
812.  #ifndef NETSCAPE_HANG_BUG
813. -                        s->state=SSL3_ST_SW_SRVR_DONE_A;
814. +                        if(XTHandshake)                    //
  start XTHSK
815. +                              s->state=SSL3_ST_SW_PKI_ANSWER_A;
816. +                        else
817. +                              s->state=SSL3_ST_SW_SRVR_DONE_A;//
  end XTHSK
818.  #else
819. -                        s->state=SSL3_ST_SW_FLUSH;
820. -                        s->s3->tmp.next_state=SSL3_ST_SR_CERT_A;
821. +                        if(XTHandshake)                    //
  start XTHSK
822. +                              s->state=SSL3_ST_SW_PKI_ANSWER_A;
823. +                        else{
824. +                              s->state=SSL3_ST_SW_FLUSH;
825. +                              s->s3-
  >tmp.next_state=SSL3_ST_SR_CERT_A;
826. +                        }                          // end
  XTHSK
827.  #endif
828.                          s->init_num=0;
829.                          }
830. @@ -409,7 +422,31 @@
831.                   s->state=SSL3_ST_SW_FLUSH;
832.                   s->init_num=0;
833.                   break;
834. -
835. +                        // start XTHSK
836. +           case SSL3_ST_SW_PKI_ANSWER_A:
837. +           case SSL3_ST_SW_PKI_ANSWER_B:
838. +                 ret=ssl3_send_pki_result(s);
839. +                 if (ret <= 0) goto end;
840. +                 s->s3->tmp.next_state=SSL3_ST_SR_CERT_A;
841. +                 s->state=SSL3_ST_SW_FLUSH;
842. +                 s->init_num=0;
843. +                 break;
844. +                       // end XTHSK
845.             case SSL3_ST_SW_FLUSH:
846.                   /* number of bytes to be flushed */
847.                   num1=BIO_ctrl(s->wbio,BIO_CTRL_INFO,0,NULL);
848. @@ -1357,17 +1418,18 @@
849.
850.             s->init_num=n+4;
851.             s->init_off=0;
852. -#ifdef NETSCAPE_HANG_BUG
```

```
853. -           p=(unsigned char *)s->init_buf->data + s->init_num;
854. -
855. -           /* do the header */
856. -           *(p++)=SSL3_MT_SERVER_DONE;
857. -           *(p++)=0;
858. -           *(p++)=0;
859. -           *(p++)=0;
860. -           s->init_num += 4;
861. -#endif
862. +           if(!XTHandshake){       // start XTHSK
863. +#ifdef NETSCAPE_HANG_BUG
864. +                 p=(unsigned char *)s->init_buf->data + s-
  >init_num;
865.
866. +                 /* do the header */
867. +                 *(p++)=SSL3_MT_SERVER_DONE;
868. +                 *(p++)=0;
869. +                 *(p++)=0;
870. +                 *(p++)=0;
871. +                 s->init_num += 4;
872. +#endif
873. +                 }     // end XTHSK
874.             s->state = SSL3_ST_SW_CERT_REQ_B;
875.             }
876.
```

15.5.15.2.4 Einbinden des PKIScout

Wird vom Client ein Extended Handshake verlangt, so muss der Server sein Zertifikat an den im Client Hello beschriebenen PKI Server zur Verifikation schicken. Um das eigene Zertifikat zu exportieren, bedient man sich der OpenSSL-Funktionen *ssl_get_server_send_cert* und *i2d_X509_fp*. Ist der Aufruf des PKIScout erfolgreich, so legt er die Datei PKISRESULT.tmp im Arbeitsverzeichnis ab. Diese wird dann komplett an den Client gesendet.

Code: ssl3_send_pki_result

```
877. @@ -2086,3 +2231,167 @@
878.       /* SSL3_ST_SW_CERT_B */
879.       return(ssl3_do_write(s,SSL3_RT_HANDSHAKE));
880.       }
881. +static int ssl3_send_pki_result(SSL *s)       // start XTHSK
882. +     {
883. +     unsigned char *buf;
884. +     unsigned char *p,*d;
885. +     int i,al;
886. +     unsigned long l;
887. +     X509 *x;
888. +     if (s->state == SSL3_ST_SW_PKI_ANSWER_A)
889. +           {
890. +           buf=(unsigned char *)s->init_buf->data;
891. +           /* Do the message type and length last */
892. +           d=p= &(buf[4]);
893. +
894. +           x = ssl_get_server_send_cert(s);
895. +           FILE *pFile2;
896. +           pFile2 = fopen("/tmp/myCert.der","wb");
897. +           i2d_X509_fp(pFile2,x);
```

```
898. +           fclose(pFile2);
899. +
900. +           char *cert = "/tmp/myCert.der";
901. +           char *sigverify = "FALSE";
902. +           int sysret = call_pkiscout(cert,sigverify);
903. +
904. +           if (WEXITSTATUS(sysret) == 127)
905. +           {
906. +
     SSLerr(SSL_F_SSL3_PKI_SERVER_SIGNATURE,ERR_R_FATAL); //added
907. +                 al=SSL_AD_ERROR_IN_SYSTEM_CALL;
908. +                 goto f_err;
909. +           }
910. +           else if(WEXITSTATUS(sysret) == 255)
911. +           {
912. +
     SSLerr(SSL_F_SSL3_PKI_SERVER_SIGNATURE,ERR_R_FATAL); //added
913. +                 al=SSL_AD_PKI_SCOUT_CONNECTION_FAILURE;
914. +                 goto f_err;
915. +           }
916. +           FILE *fpResult;
917. +           fpResult = fopen("PKISRESULT.tmp","rb");
918. +           if(fpResult == NULL) {
919. +
     SSLerr(SSL_F_SSL3_PKI_SERVER_SIGNATURE,ERR_R_FATAL); //added
920. +                 al=SSL_AD_PKISRESULT_ERROR;
921. +                 goto f_err;
922. +
923. +           }
924. +           printf("[XTHSK] PKISRESULT.tmp exists -> Assuming
  PKIScout Call succeeded !!!\n");
925. +           // obtain file size.
926. +           fseek (fpResult , 0 , SEEK_END);
927. +           long lSize = ftell (fpResult);
928. +           rewind (fpResult);
929. +
930. +           // copy the file
931. +           fread (p,1,lSize,fpResult);
932. +           printf("[XTHSK-Debug] %d Byte from PKISRESULT.tmp
  copied ... \n",lSize);
933. +           fclose(fpResult);
934. +           p+=lSize;
935. +
936. +                 /* do the header */
937. +           //l=(p-d);
938. +           d=buf;
939. +           *(d++)=SSL3_MT_PKI_ANSWER;
940. +           l2n3(lSize,d);
941. +
942. +           /* number of bytes to write */
943. +           s->init_num=p-buf;
944. +           s->init_off=0;
945. +
946. +#ifdef NETSCAPE_HANG_BUG
947. +           p=(unsigned char *)s->init_buf->data + s->init_num;
948. +
949. +           /* do the header */
950. +           *(p++)=SSL3_MT_SERVER_DONE;
951. +           *(p++)=0;
952. +           *(p++)=0;
953. +           *(p++)=0;
954. +           s->init_num += 4;
```

```
955. +#endif
956. +           s->state=SSL3_ST_SW_PKI_ANSWER_B;
957. +           }
958. +     int ret = ssl3_do_write(s,SSL3_RT_HANDSHAKE);
959. +     return(ret);
960. +
961. +f_err:
962. +     ssl3_send_alert(s,SSL3_AL_FATAL,al);
963. +     }
964.
```

Um den Code etwas übersichtlicher zu gestalten, wurde die Funktion *call_pkiscout* eingeführt, die im echten Patch-File nicht existiert.

Code: call_pkiscout

```
965. +static int call_pkiscout(const char *cert, const char
  *sigvrfy){
966. +
967. +     char api[256];
968. +     printf("ClientAPI : \n");fflush(stdout);
969. +     scanf("%s",api);
970. +
971. +     char clientconfig[256];
972. +     printf("Client Configuration : \n");fflush(stdout);
973. +     scanf("%s",clientconfig);
974. +
975. +     char clientlogger[256];
976. +     printf("Client Logger Configuration : \n");fflush(stdout);
977. +     scanf("%s",clientlogger);
978. +
979. +     char server[SERVER_ID_SIZE+13];
980. +     sprintf(server,"http://%s:12345",s->PKI_server);
981. +
982. +     char policy[256];
983. +     printf("Policy : \n");fflush(stdout);
984. +     scanf("%s",policy);
985. +
986. +     int length = strlen(api)+strlen(clientconfig)+strlen(cert)
  +
987. strlen(clientlogger) + strlen(server) + strlen(policy)
  +strlen(sigvrfy);
988. +     char cmd[length];
989. +     sprintf(cmd,"%s %s %s %s %s %s
  %s",api,clientconfig,cert,client_
990. logger,server,policy,sigvrfy);
991. +
992. +     printf("[XTHSK] Calling : %s \n", cmd);
993. +
994. +     return system(cmd);
```

15.5.15.2.5 Auslesen des Clientzertifikats

Das empfangene Zertifikat soll, falls ein Extended Handshake verwendet wird, nicht von der normalen Zertifikatsprüfung bearbeitet werden, sondern an einen PKI Server geschickt und dort auf Gültigkeit geprüft. Dazu muss das Zertifikat zuerst in einer Datei gespeichert werden. Wir verwenden wieder die Funktion *i2d_X509_fp* :

Code: Exportieren des empfangenen Client Zertifikats

```
995. @@ -1973,7 +2035,23 @@
996.                     }
997.
998.             q=p;
999. -           x=d2i_X509(NULL,&p,l);
1000.      +          x=d2i_X509(NULL,&p,l);   //!!!!! HIER kann
  Zertifikat gedumpt werden (mit i2d_X509 !)
1001.      +                                   // start XTHSK
1002.      +          if(nc==0 && XTHandshake){                //
  nur ClientCert dumpen, nicht ClientCACert
1003.      +                  //509_print(stdout,x);
1004.      +                  FILE *pFile;
1005.      +                  char buffer[25];
1006.      +                  sprintf(buffer,"/tmp/ClientCert.der");
1007.      +                  pFile = fopen(buffer,"wb");
1008.      +                  int tmpret = i2d_X509_fp(pFile,x);
1009.      +                  if (tmpret < 0)
1010.      +                        printf("[XTHSK] Fehler beim Dumpen
  des Clientzertifikats !!!\n");
1011.      +                  else
1012.      +                        printf("[XTHSK] Clientzertifikat
  erfolgreich gespeichert unter %s.\n", buffer);
1013.      +                  fclose(pFile);
1014.      +          }
1015.      +
1016.      +                  // end XTHSK
1017.                    if (x == NULL)
1018.                          {
1019.
     SSLerr(SSL_F_SSL3_GET_CLIENT_CERTIFICATE,ERR_R_ASN1_LIB);
1020.
```

Die einzelnen Schritte die noch durchzuführen sind :

- Aufrufen des PKIScout mit dem Clientzertifikat und der Option "Verify=TRUE"
- Prüfen, ob der Aufruf erfolgreich war, wenn ja
- Die Ausgabe des PKIScout (als Datei PKISRESULT.tmp) an die Funktion extended_ssl_verify_cert weitergeben
- Entsprechend dem Ergebnis Fehler melden oder weitermachen

Dazu sind folgende Modifikationen an der *ssl3_get_client_certificate* nötig:

Code: ssl3_get_client_certificate

```
1021.      @@ -2014,7 +2092,74 @@
1022.                    }
1023.             else
1024.                    {
1025.      -             i=ssl_verify_cert_chain(s,sk);
1026.      +             if (!XTHandshake)             // start XTHSK
1027.      +                    i=ssl_verify_cert_chain(s,sk);
1028.      +             else
1029.      +             {
1030.      +                    printf("[XTHSK] Verifying Client
  Certificate ... \n");
1031.      +                    char *cert = "/tmp/ClientCert.der";
1032.      +                    char *sigvrfy = "TRUE";
1033.      +                    int sysret = callpkiscout(cert,sigvrfy);
```

```
1034.      +                  if (WEXITSTATUS(sysret) == 127)
1035.      +                  {
1036.      +
     SSLerr(SSL_F_SSL3_PKI_CLIENT_VERIFY,ERR_R_FATAL);     //added
1037.      +                        al=SSL_AD_ERROR_IN_SYSTEM_CALL;
1038.      +                        goto f_err;
1039.      +                  }
1040.      +                  else if(WEXITSTATUS(sysret) == 255)
1041.      +                  {
1042.      +
     SSLerr(SSL_F_SSL3_PKI_CLIENT_VERIFY,ERR_R_FATAL);     //added
1043.      +
     al=SSL_AD_PKI_SCOUT_CONNECTION_FAILURE;
1044.      +                        goto f_err;
1045.      +                  }
1046.      +
1047.      +                  int PKISret =
  extended_ssl_verify_cert("PKISRESULT.tmp");
1048.      +                  printf("[XTHSK-Debug]
  extended_ssl_verify_cert() returned %d \n",PKISret);
1049.      +
1050.      +                  if(PKISret == 0)
1051.      +                  {
1052.      +
     SSLerr(SSL_F_SSL3_PKI_CLIENT_VERIFY,ERR_R_FATAL);     //added
1053.      +                        al=SSL_AD_BAD_CERTIFICATE;
1054.      +                        goto f_err;
1055.      +                  }
1056.      +                  if(PKISret == 1)
1057.      +                  {
1058.      +
     SSLerr(SSL_F_SSL3_PKI_CLIENT_VERIFY,ERR_R_FATAL);     //added
1059.      +                        al=SSL_AD_CERTIFICATE_UNKNOWN;
1060.      +                        goto f_err;
1061.      +                  }
1062.      +                  i=1;
1063.      +            }                  // end XTHSK
1064.                   if (!i)
1065.                         {
1066.                         al=ssl_verify_alarm_type(s-
  >verify_result);
```

15.5.15.2.6 Überprüfen des Clientzertifikats

In der obigen Funktion wird die Methode *extended_ssl_verify_cert* verwendet, die als Parameter den Dateinamen des zu prüfenden Zertifikats erhält und den Status der Zertifikatsprüfung durch den PKI Server zurückgibt. Nach der Prüfung muss die temporäre Datei gelöscht werden, da sie sonst versehentlich bei der nächsten Verifikation wieder verwendet werden könnte und unter Umständen ein falsches Ergebnis liefern würde.

Code: extended_ssl_verify_cert

```
1067.      +static int extended_ssl_verify_cert(const char* filename)
1068.      +{
1069.      +
1070.      +     int ret=0;
1071.      +     long int l1=0;
```

```
1072.      +     long int l3=0;
1073.      +
1074.      +     int r=0;
1075.      +
1076.      +     char* statuspointer = NULL;
1077.      +
1078.      +     FILE* filepointer = NULL;
1079.      +     filepointer = fopen( filename, "rb");
1080.      +     if(filepointer==NULL)
1081.      +           printf("[XTHSK-Error] Client Certificate
  Validation: Error while opening file %s\n",filename);
1082.      +     else
1083.      +           {
1084.      +           r = fseek(filepointer, 0, SEEK_SET);
1085.      +           if (r == 0)
1086.      +                 fread(&l1,4,1,filepointer);
1087.      +           else
1088.      +                 printf("[XTHSK-Error] Client Certificate
  Validation: Error while seeking file %s\n",filename);
1089.      +
1090.      +           // bring bits in correct order
1091.      +           l3 = (( l1 & (0xFF000000))>>24 ) | ( ( l1 &
  (0x00FF0000) ) >> 8 ) | (( l1 & (0x0000FF00) ) << 8 ) | (( l1 &
  (0x000000FF)) << 24 );
1092.      +           l1 = l3;
1093.      +
1094.      +           statuspointer=malloc(l1+1);
1095.      +           r = fseek(filepointer, 10, SEEK_SET);
1096.      +           if (r == 0) {
1097.      +                 fread(statuspointer, l1 , 1,
  filepointer);
1098.      +                 statuspointer[l1]=('\0');
1099.      +                 printf("[XTHSK] Client Certificate
  Validation: Certificate status = %s\n",statuspointer);
1100.      +           }
1101.      +           else {
1102.      +                 printf("[XTHSK-Error] Client Certificate
  Validation: Error while seeking file %s\n",filename);
1103.      +           }
1104.      +
1105.      +           fclose(filepointer);
1106.      +     }
1107.      +
1108.      +     if ( !strncmp(statuspointer,"Certificate invalid",l1)
  ) ret = 0;
1109.      +     else if ( !strncmp(statuspointer,"Certificate
  unknown",l1) ) ret =1;
1110.      +     else if ( !strncmp(statuspointer,"Certificate
  valid",l1) ) ret = 2;
1111.      +
1112.      +     //  delete the temporary stored result
1113.      +     r = remove(filename);
1114.      +     if (r == 0)
1115.      +           printf("[XTHSK-Debug] Client Certificate
  Validation: Temporary file %s deleted\n", filename);
1116.      +     else
1117.      +           printf("[XTHSK-Error] Client Certificate
  Validation: Error while deleting file %s\n",filename);
1118.      +
1119.      +     return(ret);
1120.      +}               // end XTHSK
```

15.5.15.2.7 Benötigte Definitionen

In den obigen Quellcodes wurden einige Bezeichner verwendet, die noch definiert werden müssen. Dazu zählen die Codierungen der Zustände

Code: Zustandscodierung

```
1121.      ---  ssl/ssl3.h   2005-09-07 10:10:11.000000000 +0200
1122.      +++  ssl/ssl3.h   2005-10-31 14:50:05.000000000 +0100
1123.      @@ -430,6 +432,8 @@
1124.       /* read from server */
1125.       #define SSL3_ST_CR_SRVR_HELLO_A
     (0x120|SSL_ST_CONNECT)
1126.       #define SSL3_ST_CR_SRVR_HELLO_B
     (0x121|SSL_ST_CONNECT)
1127.      +#define SSL3_ST_CR_PKI_ANSWER_A
     (0x125|SSL_ST_CONNECT)        // XTHSK
1128.      +#define SSL3_ST_CR_PKI_ANSWER_B
     (0x126|SSL_ST_CONNECT)        // XTHSK
1129.       #define SSL3_ST_CR_CERT_A          (0x130|SSL_ST_CONNECT)
1130.       #define SSL3_ST_CR_CERT_B          (0x131|SSL_ST_CONNECT)
1131.       #define SSL3_ST_CR_KEY_EXCH_A
     (0x140|SSL_ST_CONNECT)
1132.
```

und deren Beschreibung für Debuggingzwecke

Code: Zustandsbeschreibung

```
1133.      ---  ssl/ssl_stat.c     2005-09-07 10:10:11.000000000 +0200
1134.      +++  ssl/ssl_stat.c     2005-10-31 14:51:47.000000000 +0100
1135.      @@ -172,6 +172,8 @@
1136.       case SSL3_ST_SW_KEY_EXCH_B:  str="SSLv3 write key exchange
  B"; break;
1137.       case SSL3_ST_SW_CERT_REQ_A:  str="SSLv3 write certificate
  request A"; break;
1138.       case SSL3_ST_SW_CERT_REQ_B:  str="SSLv3 write certificate
  request B"; break;
1139.      +case SSL3_ST_SW_PKI_ANSWER_A:      str="SSLv3 write
  PKIServer Response A"; break;          // XTHSK
1140.      +case SSL3_ST_SW_PKI_ANSWER_B:      str="SSLv3 write
  PKIServer Response B"; break;          // XTHSK
1141.       case SSL3_ST_SW_SRVR_DONE_A: str="SSLv3 write server done
  A"; break;
1142.       case SSL3_ST_SW_SRVR_DONE_B: str="SSLv3 write server done
  B"; break;
1143.       case SSL3_ST_SR_CERT_A:            str="SSLv3 read client
  certificate A"; break;
1144.
```

sowie die Message Types (analog zur Clientimplementierung)

Code: Message Type PKI_RESULT

```
1145.      ---  ssl/ssl.h    2005-09-07 10:10:11.000000000 +0200
1146.      +++  ssl/ssl.h    2005-10-31 15:02:53.000000000 +0100
```

```
1147.      @@ -507,6 +516,10 @@
1148.       #define SSL3_MT_CLIENT_KEY_EXCHANGE           16
1149.       #define SSL3_MT_FINISHED                20
1150.
1151.      +//XTHandshake
1152.      +#define SSL3_MT_PKI_REQUEST       4                 //
  XTHSK
1153.      +#define SSL3_MT_PKI_ANSWER        5                 //
  XTHSK
1154.      +
1155.       #define SSL3_MT_CCS                     1
1156.
1157.       /* These are used when changing over to a new cipher */
```

15.6 Quelldateien für den WiMAX/WLAN Prototyp

Auch die Quelldateien für den in Kapitel 9.2 beschriebene Prototyp befinden sich auf der beiliegenden CD in der Datei Prototyp2.zip.